A CAN Transceiver for a Smart Output ASIC of Automotive Electronic Control Units: Design, Implementation, and Measurement

Won-Hee Jo

Department of Electrical Engineering

Graduate School of UNIST

2016

A CAN Transceiver for a Smart Output ASIC of Automotive Electronic Control Units: Design, Implementation, and Measurement

Won-Hee Jo

Department of Electrical Engineering

Graduate School of UNIST

A CAN Transceiver for a Smart Output ASIC of Automotive Electronic Control Units: Design, Implementation, and Measurement

A thesis
submitted to the Department of Electrical Engineering
in partial fulfillment of the
requirements for the degree of
Master of Science

Won-Hee Jo

Jun. 14, 2016

Approved by

Advisor
Myunghee Lee

A CAN Transceiver for a Smart Output ASIC of Automotive Electronic Control Units: Design, Implementation, and Measurement

Won-Hee Jo

This certifies that the thesis of Won-Hee Jo is approved.

Jun. 14, 2016

Advisor: Myunghee Lee

Franklin Bien

Seokhyeong Kang

ABSTRACT

In this thesis, the design, implementation, and measurement of a CAN transceiver are presented as a part of an architecture of Multi-domain Electronic Control Unit (MECU) and Smart Output ASIC for better mileage, wire length reduction and flexible system configuration. The proposed Smart Output ASIC is a dedicated terminal system for actuation feature. The main component includes MCU core, four types of actuator drivers, CAN/LIN transceiver, temperature sensor, and BGR/LDO regulator.

The Smart Output ASIC communicates with a central ECU by using CAN or LIN protocols. Therefore, the first step for the Smart Output ASIC is to implement the CAN transceiver meeting ISO 11898-2/-5. In this work, two types of CAN transceiver were implemented; one is a stand-alone version CAN transceiver (Type I), and another is an MCU integrated version (Type II). Both support high-speed CAN which has the maximum data rate of 1Mbps. They are thermally protected, and robust against high voltage hazard. An 8-bit MCU is used for Type II. Type II is the first prototype of Smart Output ASIC, and it would be extended to the complete Smart Output ASIC by embedding the actuator drivers. The designed CAN transceivers were fabricated by using automotive 0.18um, 52V BCDMOS technology. The chip size of the Type I is 1,275um (W) × 1,125um (H) except I/O pads. Also, Type II was implemented with the size of 2,680um (W) × 3,280um (H) except I/O pads. The fabricated CAN transceivers were tested for verifying the interoperability with commercial products.

KEYWORD: Multi-domain, Smart Output ASIC, Controller Area Network, MCU integrated CAN, high voltage protection, thermal protection, fail-safe, fault tolerant

CONTENTS

LIST OF FIGURE

LIST OF TABLE

NOMENCLATURE

ACK	Acknowledgement
ADC	Analog to Digital Converter
ASIC	Application Specific Integrated Circuit
BCDMOS	Bipolar Complementary Double-diffused MOS
BGR	Bandgap Reference
CAN	Controller Area Network
CPU	Central Processing Unit
DAC	Digital to Analog Converter
DDM	Driver-side Door-zone Module
DMOS	Double-diffused MOS
ECU	Electronic Control Unit
EDMOS	Extended Drain MOS
EEPROM	Electrically Erasable Programmable ROM
EMI	Electromagnetic Interference
EME	Electromagnetic Emission
EMC	Electromagnetic Compatibility
ESD	Electrostatic Discharge
FSM	Finite State Machine
GPIO	General Purpose Input and Output
HSD	High-side Driver
HSS	High-side Switch
I/O	Input / Output
ISO	International Organization of Standardization
KATECH	Korea Automotive Technology Institute

LDO	Low-drop Out
LDMOS	Laterally Diffused MOS
LED	Light Emitting Diode
LIN	Local Interconnect Network
LSD	Low-side Driver
LSS	Low-side Switch
LQFP	Low-profile Quad Flat Package
MCU	Micro-Controller Unit
MECU	Multi-domain ECU
MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistor
MUX	Multiplexer
P&R	Place & Route
PCB	Printed Circuit Board
PMIC	Power Management Integrated Circuit
PoR	Power-on Rest
PVT	Process, Voltage, and Temperature
PWM	Pulse Width Modulation
RAM	Random Access Memory
ROM	Read Only Memory
RTL	Register Transfer Level
RXD	Receive Data
SoC	System on Chip
TXD	Transmit Data

I. Introduction

In automotive industry, achieving the better fuel efficiency is age-old issue, and has always been a challenge to automotive manufacturers [1]. As electronic technology has been grown, the car makers have employed more electronic systems for infotainment, passenger convenient features, chassis control, and even powertrain control. Fig. 1.1 shows the electronics innovations in automotive industry through the years [2].

Nowadays, a high-end passenger car has 70 to 100 microprocessor-based ECUs with network, and even low-end cars have 30 to 50 MCUs [3]. According to Infineon's estimation, electronics in a car is estimated to increase to 35% within the next 6 years [2]. It means the weight of electronics and harness in a car has been increased, and will be increased. There is a classic case to reduce the wire harness. In 1980s, Bosch tried to reduce the wire harnesses by inventing a CAN bus [4]. In spite of this effort, the weight of wire harness has been steadily increasing, and today's car have more than 4km of wiring [5].

The inefficient architecture of electronic system induces the messy and complex wiring, and this inefficient wire network leads to increase the weight of harness. It means poor mileage, rise of production cost, weakening of reliability, and limiting expanded functionality. Fig. 1.2 shows an example of an electronic system structure of a recent passenger car, which is very complicated wire harnesses.

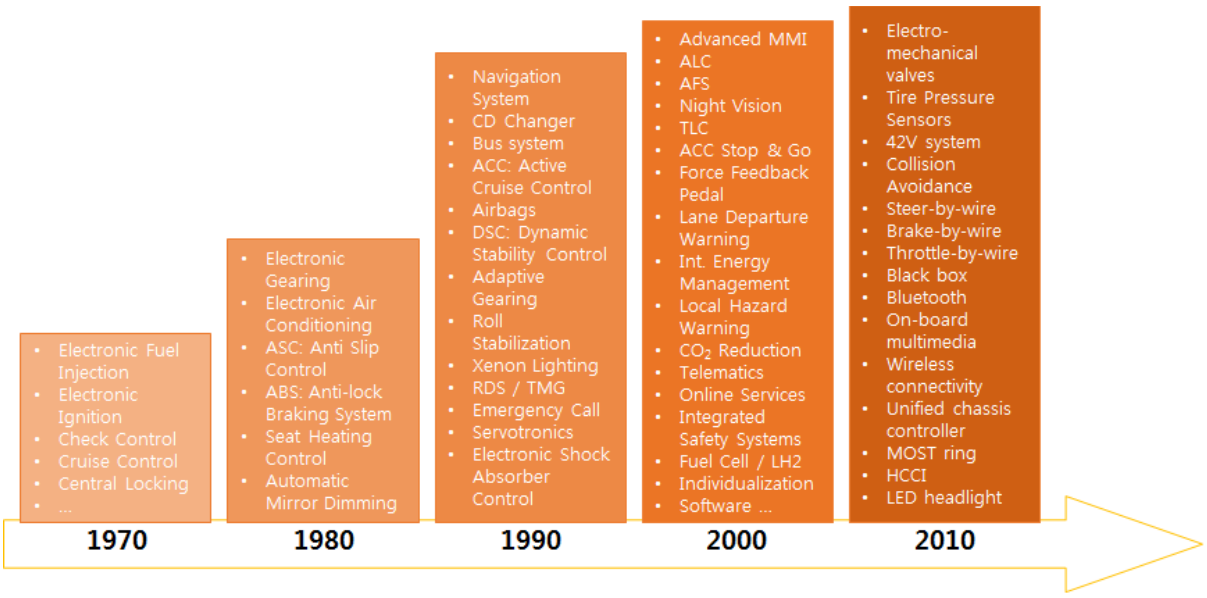


Figure 1.1 Automotive Electronics innovations through the years [2]

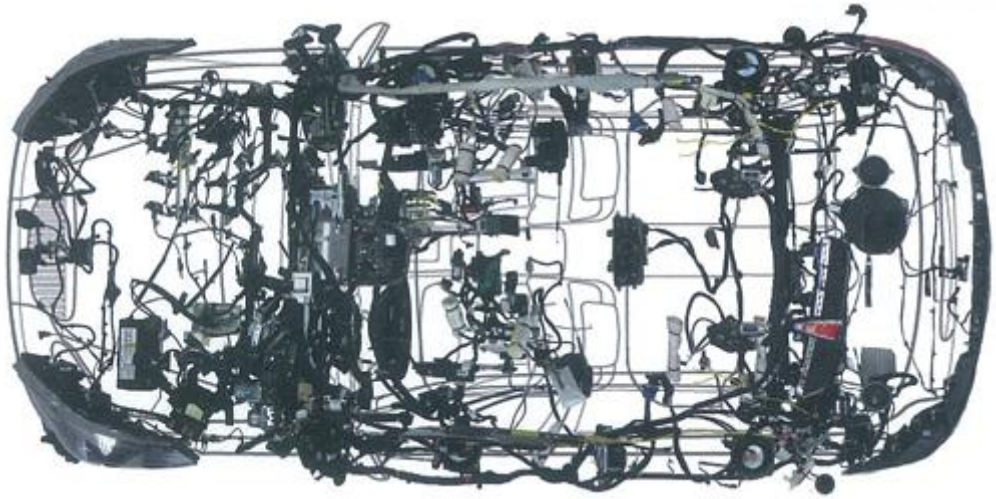


Figure 1.2 An example of today's automotive electronic system and wiring network (image source: Wise Automotive, http://www.wise-automotive.kr)

The structural complexity is very high because the system is implemented based on lots of heterogeneous ECUs. Currently, more and more ECUs are adopted in a car for providing more functionalities, but the functionalities have been implemented by just adding more ECUs in a car. The result is shown in Fig. 1.2.

A number of various ECUs creates several issues on the system. First, the conventional architecture like Fig. 1.2 induces the difficulty of manufacturing. Second, the new function has to be implemented by adding a new ECU, and it requires the large resource of development. A totally different functionality of ECU should be implemented from hardware level. Besides, it must meet reliability specification, and be compatible with pre-installed system. Third, the maintenance cost would be increased due to high system complexity. Fourth, the weight of wiring would be increased because of the dense harness.

II. Smart Output ASIC

The conventional system architecture has many issues as reviewed in Section I. In this work, Multi-domain ECU (MECU) architecture is proposed to manage these issues. MECU is a unified automotive ECU for future vehicles. It unifies the various functionalities of ECUs in vehicle for lightening. Fig. 2.1 presents the concept of MECU.

An architecture (a) of Fig. 2.1 is current architecture. The ECUs are used only for a specific task. There are redundant MCUs. Then, it is connected to sensors and actuators directly, which create a complicated wiring network. The second architecture (b) 'integrate all things of SoC' version. But it does not help to reduce the harnesses. Finally, the architecture (c) is MECU-based. It centralized the functionalities of logical decision. Then, the functionalities of sensors, switches, motor/lamp actuators are detached from the central ECU and distributed for localization. Definitely, it greatly helps to reduce the number of wire harness.

The detailed concept of MECU architecture is presented in Fig. 2.2. The input logics of each ECU are combined to a single functional node which called "Smart Input ASIC." The output logic of each ECU are also detached and combined to "Smart Output ASIC." Many redundant small MCUs are unified to a powerful central application processor, i.e. MECU. This is the concept of MECU architecture. There are three main components; MECU, Smart Input ASIC, and Smart Output ASIC.

MECU is a real-time multi-domain ECU which unifies all of the logical functionality from respective domain's ECUs. It does not divide the system domains anymore, such as power train, body, chassis, and infotainment. Since the distributed system becomes a centralized system, MECU must have an enough computation capability to process the data of whole system.

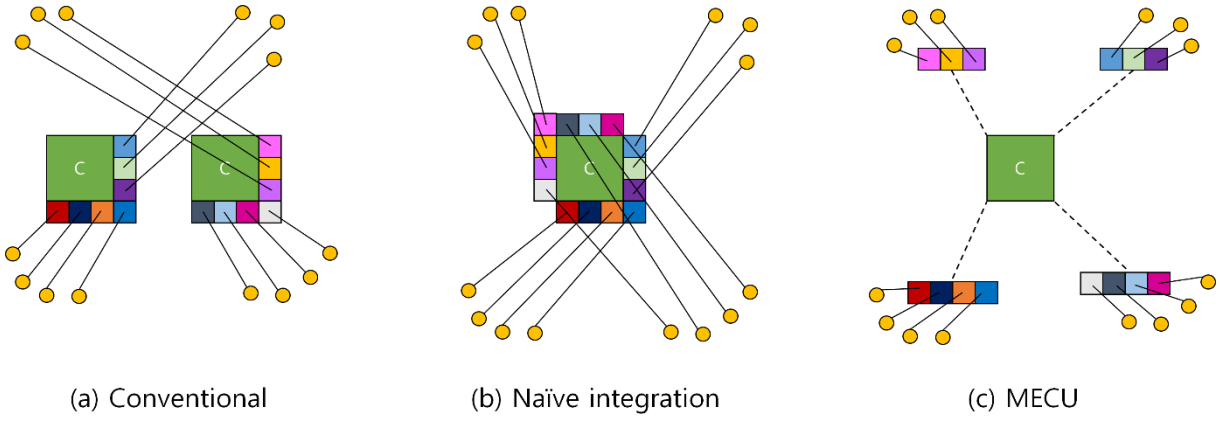


Figure 2.1 Automotive electronic system architectures: (a) conventional, (b) naïve integration (c) proposed MECU

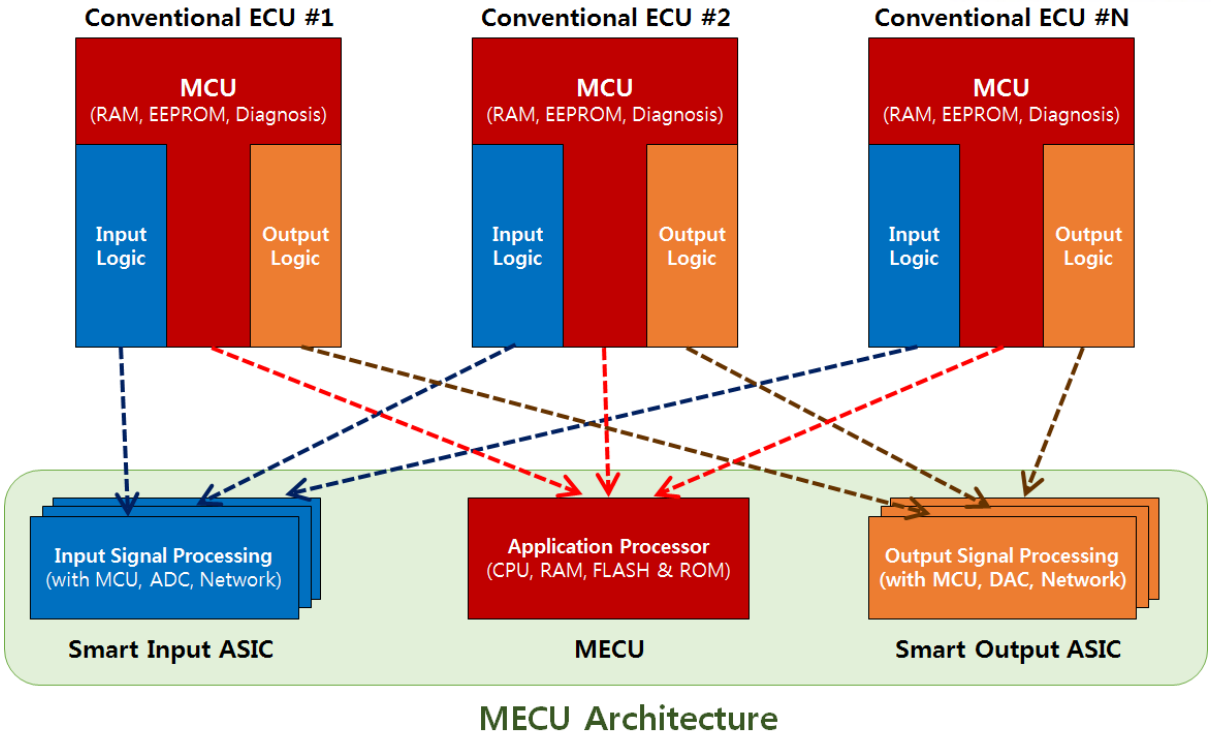


Figure 2.2 Concept of MECU architecture

Smart Input/Output ASIC is a terminal system. It is a dedicated small system for sensing and actuation function. It pre-processes the input data coming from various sensors. Then, transmit the preprocessed data to MECU. Fig. 2.3 describes this concept.

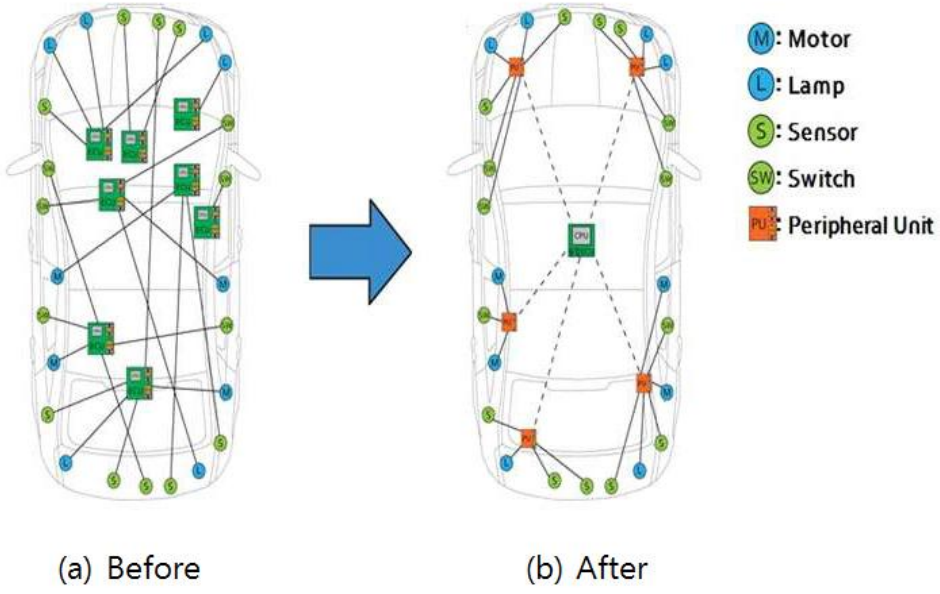


Figure 2.3 (a) current system network, (b) after applying MECU architecture.

(image source: Wise Automotive, http://www.wise-automotive.kr)

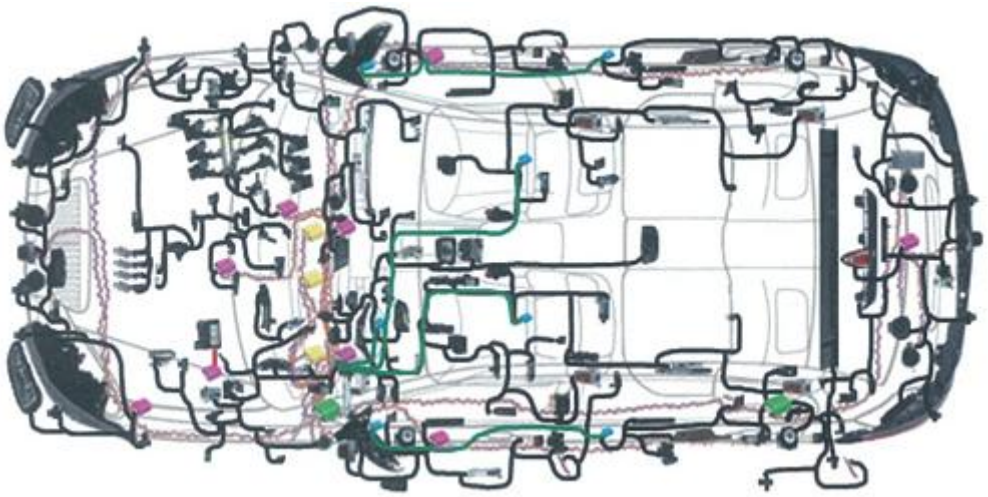


Figure 2.4 An example of future automotive electronic system and wire net

(image source: Wise Automotive, http://www.wise-automotive.kr)

This MECU architecture offers following advantages. First, whole automotive electronic system could be implemented by fewer ECUs. Second, it is easy to implement new features. MECU can be re-programmed anytime, thus engineers can add new features by just updating software. Third, maintenance cost would be reduced because of less wire harness and ECUs. Fourth, the weight of wiring would be decreased thanks to shorter wires. Consequently, future automotive electronic system and wiring net would be like Fig. 2.4 by using proposed MECU architecture.

Fig. 2.5 shows the roles in system. Smart Input ASIC accepts the signal from switches and sensors.

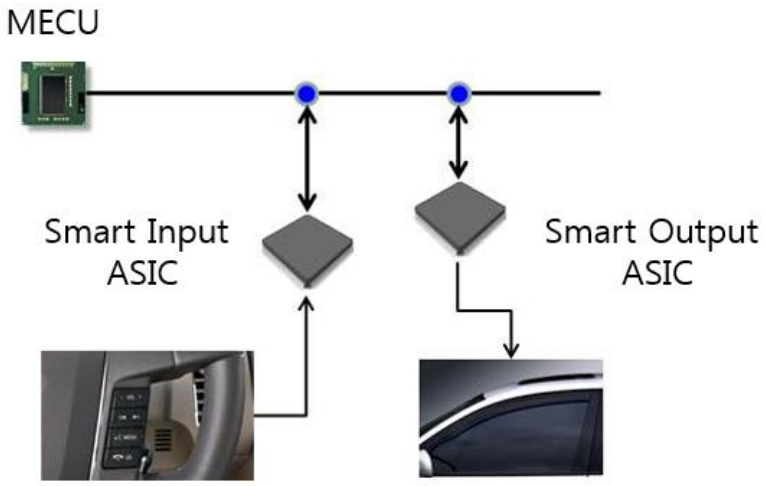


Figure 2.5 Concept of Smart Input ASIC and Smart Output ASIC

The input signals come from GPIO, ADC, PWM, and so on. Smart Input ASIC digitalizes these analog signals and preprocesses it before it sends the preprocessed data to MECU by using network such as CAN protocols. MECU collects overall status of a car by using incoming data. MECU processed the data, and send a command to a proper Smart Output ASIC node through CAN bus for necessary actions.

The Smart Output ASIC supports various configurations for driving actuators; full-bridge, half-bridge, high-side, and low-side. The actuator configuration can be modified by just MCU software. It means there is no need to redesign the hardware. It needs just programming for handling another application. This is the reason why the Smart Output ASIC is called as 'Smart.'

Fig. 2.6 (a), (b) shows their architecture, Smart Input ASIC and Smart Output ASIC respectively. Both of Smart Input ASIC and Smart Output ASIC have their internal MCU, thus the configuration is flexible and easy to modify. Also, they can diagnose themselves, and they own their network, so they can send a report to MECU.

This thesis covers the design and implementation of a CAN transceiver for this Smart Output ASIC. The following section covers the design issues on a CAN transceiver.

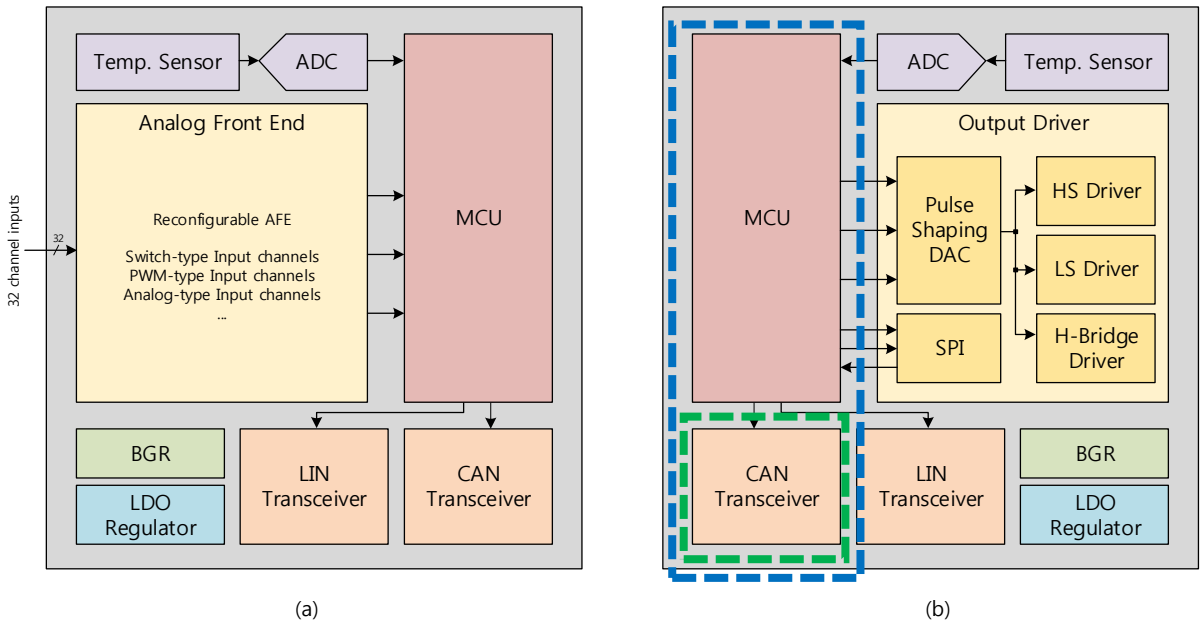


Figure 2.6 (a) Architecture of Smart Input ASIC, (b) Architecture of Smart Output ASIC

III. Design, Implementation, and Measurement of a CAN Transceiver

3.1 Design Considerations

This section briefly introduces automotive environments and provides design considerations for automotive ICs. The automotive vehicles are directly exposed to the natural environment, and the operating conditions are very tough for electronic devices. These conditions should be carefully reviewed when designing an IC for automotive applications.

As mentioned, some of the automotive environment is harsh to electronics. Those includes a wide range of temperature change, disastrous power supply, mechanical vibration, fluidic exposure, EMI/ESD and guaranteed long life for more than 10 years. The detailed items are listed in Table 3.1. Among many kinds of hazards, mainly focused item would be temperature and power supply. Because these are key factors in design process, and other items are more likely to be related with reliability issue, not with electrical design.

First, the operating temperature range for automotive is from -40℃ to 165℃. The characteristic of semiconductor would be changed along the temperature change. In extreme cases, it causes a failure of IC. Considering the most consumer ICs have to meet the temperature condition from 0℃ to 85℃, this would be a big difference between consumer and automotive application. Fig. 3.1 presents one of example for the thermal environment of engine room. For this reason, the commercial CAN transceivers support the operating temperature up to 150℃.

Table 3.1 The automotive environment [6]

Temperature	Driver interior	-40℃ to +85℃
	Under hood	-40℃ to +125℃
	On-engine	-40℃ to +150℃
	In the exhaust and combustion areas	-40℃ to +200-600℃
Mechanical Shock	During assembly (drop test)	3000g
	On the vehicle	50-500g
Mechanical Vibration		15g, 100Hz to 2kHz
Electromagnetic Impulses		100 to 200V/m
Exposure to	Common	Humidity, salt spray
	In some application	Fuel, oil, brake fluid, transmission fluid, ethylene glycol, exhaust gases

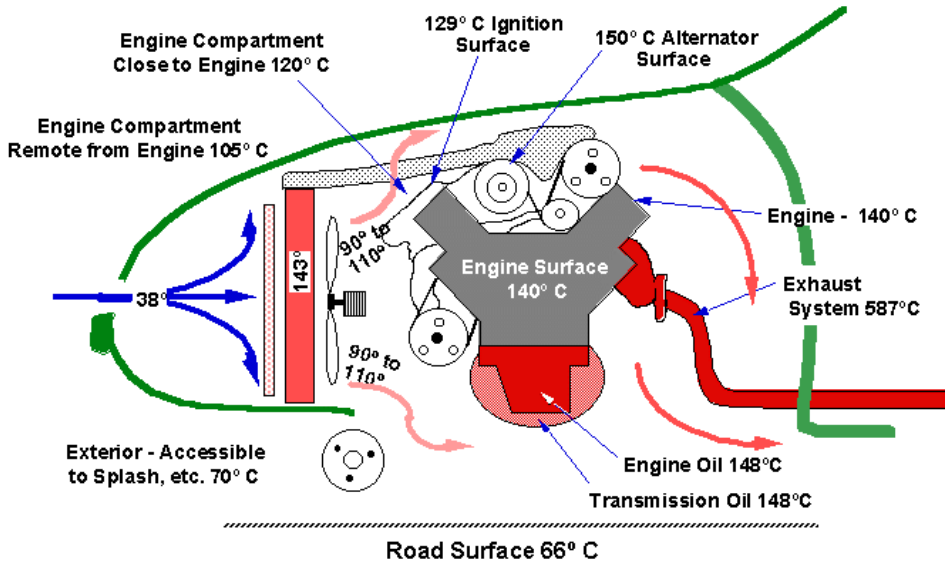


Figure 3.1 The thermal profile of an engine compartment [7]

The power supply of automotive is very noisy and highly volatile. Fig. 3.2 shows a typical power supply chain of a car. There is only one power supply potential, V_{BAT}, the battery voltage. The battery is a main power source when the engine is off while the alternator generates the power when the engine is running. The nominal voltage level of a car battery is typically 12.6V, but it could drop to 8V when the battery is almost discharged. On the other hands, it could go up to 18V when battery is charging by using the alternator. Since the battery voltage is fluctuating with wide range, an automotive ICs could be operating under this wide range of the power supply voltage.

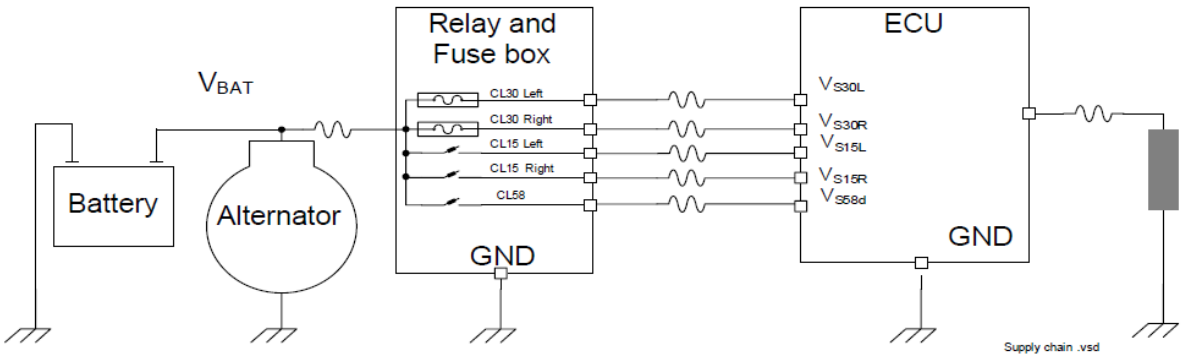


Figure 3.2. Typical supply chain in a vehicle [8]

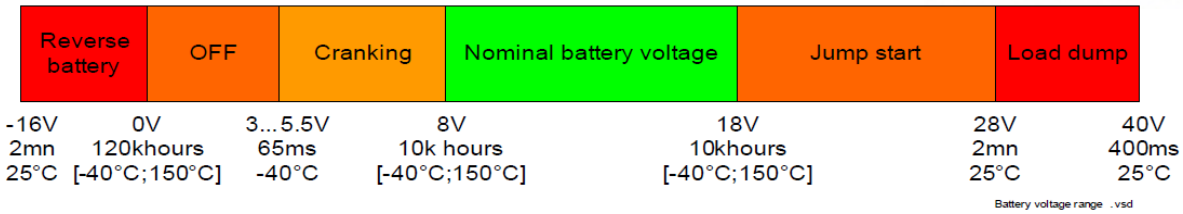


Figure 3.3 A typical specification for power supply voltage inside a car [8]

Fig. 3.3 shows the variation of power supply voltage level [8]. For more details about supply voltage range, refer to [8]. Therefore, the standard of CAN, especially ISO 11898-5 suggests the maximum ratings of CANH, CANL, and V_{Split} as described in Table 3.2.

Table 3.2 Maximum ratings of CANH, CANL, and VSplit [4]

Nominal batter voltage (V)	Notation	Voltage	
		Vmin	Vmax
14	VCAN_H	-27.0	+40.0
	VCAN_L	-27.0	+40.0
	VSplit	-27.0	+40.0

3.2 CAN Transceiver Type I: Stand-alone Version

3.2.1 Architecture

The proposed CAN transceiver is designed based on ISO 11898-2 and ISO 11898-5 supporting data rate of 1 Mbps [4]. The CAN transceiver supports a low-power mode, and remote wake-up feature which is described in ISO 11898-5. The transceiver disengages from the bus when it is in unpowered status. It included an integrated 5V-to-3V LDO, in order to support both 3.3V and 5V for I/O interface. It provides a voltage source (V_{SPLIT}) for stabilizing the recessive bus level for further improvement of EME. The pins (CANH, CANL, and V_{SPLIT}), which connected to CAN bus outside the IC, should be protected against high voltage transient in automotive environment: from -27V to 40V. The transceiver also should thermally be protected. It provides the fail-safe function such as a TXD dominant time out. A block diagram of the proposed CAN transceiver is shown in Fig. 3.4, and the key specification is described in Table 3.3.

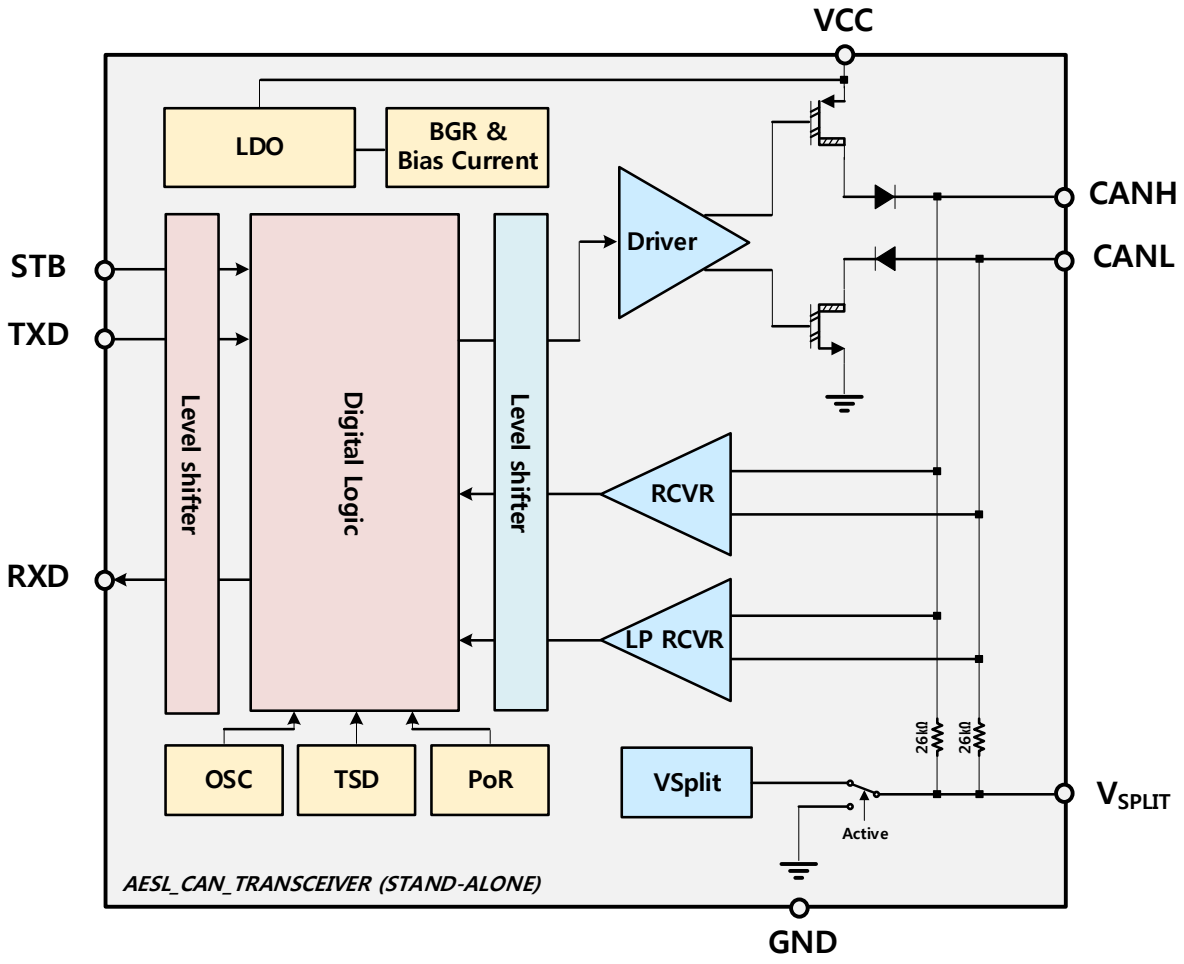


Figure 3.4 Block diagram of proposed CAN transceiver

Table 3.3 Summarized specification of implemented CAN transceiver

Symbol	Description	Min	Typ	Max	Unit
V_{CC}	Supply voltage (operating range)	4.75	5	5.25	V
I_{CC}	Supply current (transit dominant bit)	30	43	70	mA
V_{CANH}	Output voltage at pin V_{CANH}*	3.24	3.5	3.77	V
V_{CANL}	Output voltage at pin V_{CANL}*	1.39	1.5	1.7	V
V_{SPLIT}	Output voltage at pin V_{SPLIT} (unloaded)*	2.36	2.48	2.6	V
V_{ref}	Reference voltage*	1.88	1.95	2.02	V
f_{op}	Operating frequency*	2	4	6	MHz
t_{rst}	Power-on reset pulse duration*	291	2,470	160,000	us
t_{TXD_RXD}	Propagation delay TXD to RXD	170	-	203	ns

* Simulated with voltage variation of $\pm 5\%$, and process variation (ff/tt/ss)

In Fig. 3.4, red colored boxes mean 1.8V-domain modules, yellow boxes are 3.3V voltage domain, and blue ones are 5V voltage domain. The main components are ***Driver***, 'protection diodes and switch MOS', and '***RCVR*** (receiver)'. Both form the signal path from TXD to RXD. ***LP RCVR*** is a dedicated block for sensing a remote wake-up signal from the bus during low-power mode. ***OSC*** block generates the clock signal for ***Digital Logic*** block. ***TSD*** block senses the over-temperature, and shut down the device in order to prevent abnormal operations. Power-on-Reset (***PoR)*** block provides a reset pulse for initializing the ***Digital Logic*** block when power is on. ***VSplit*** is a voltage source, and it reduces EME.

3.2.2 Design and Implementation

3.2.2.1 Protection diodes and switch LDMOS

There are two protection diodes in signal path, and they protect the transceiver from high voltage of +40V or reverse voltage of –16V (-27V in ISO 11898-5). This specification is described in ISO 11898-2, and it is mentioned how the voltage levels are determined in Section 3.1. CAN bus is a long twisted wire pair. Hence, the shied of the bus cable might be damaged due to repetitive folding and unfolding at door hinge or some physical impulse shock or mechanical vibration. Since the chassis of a car acts as a ground terminal, the peeled wire might be shorted to power chain of battery. As mentioned in Section 3.1, during 'Load dump', the supply voltage goes up to 40V and the peeled wire might be shorted to chassis. For preventing the device from this high voltage hazard, protection diodes are needed. In this design, 52V Schottky diodes were selected for the high voltage protection.

Two DMOSs are the main switches, and they also have to endure against the high voltage. When a reversed voltage is applied to CAN bus, the DMOS should not be damaged. The diodes do not help, since it is in forward bias. Therefore, DMOS has to withstand the voltage of -16V. The diodes and DMOSs occupy the most of area as shown in Fig. 3.7. The current in the bus line is typically around of 45mA. Hence, the size of switch DMOSs and diodes are sufficiently large for the current driving capability. Most case, LDMOS is chosen for switching functionality, but in this work, EDMOS was used due to limitation of foundry process.

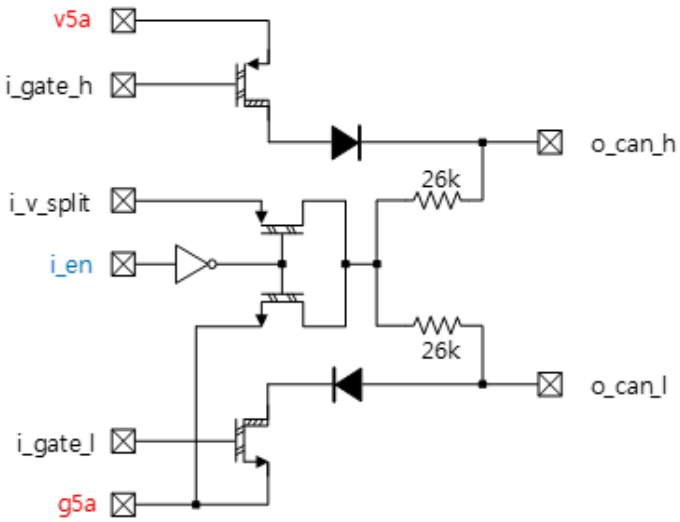


Figure 3.5 The schematic of *BI* block

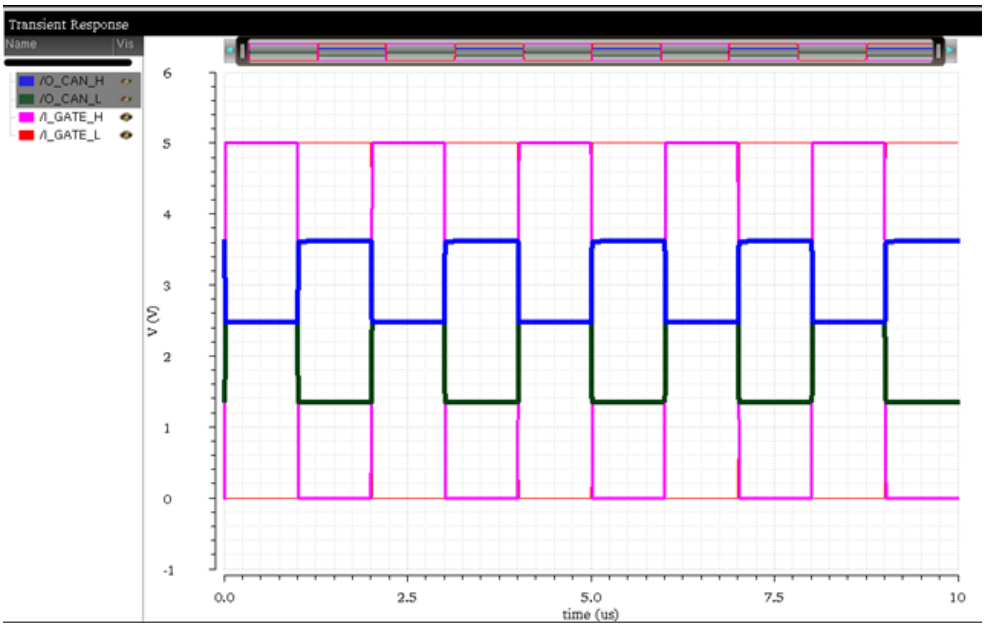


Figure 3.6 Simulation result of *BI* block

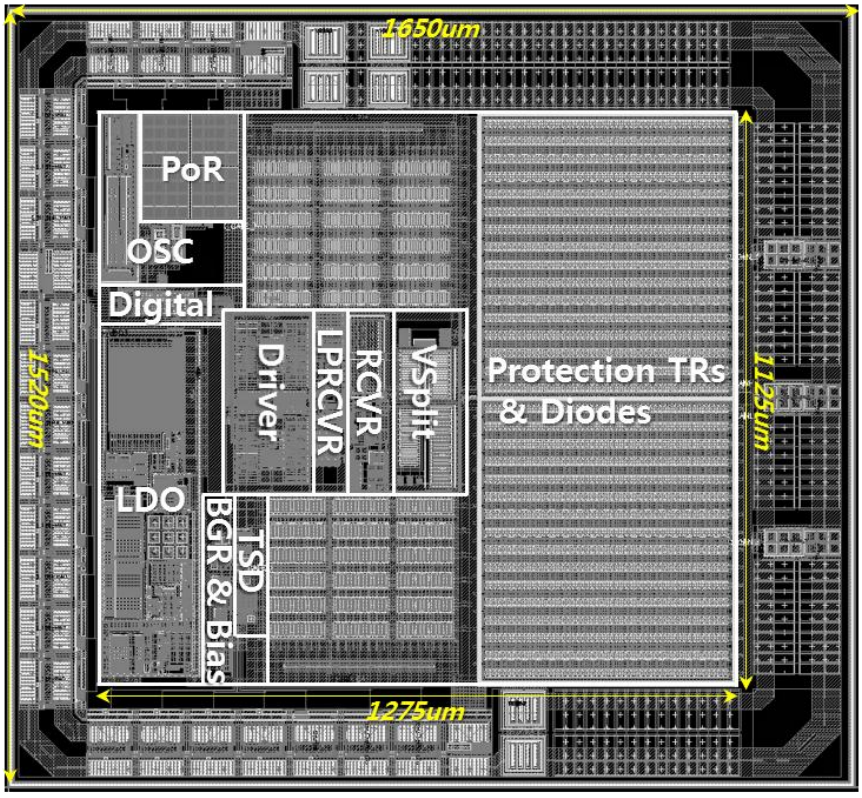


Figure 3.7 Layout of CAN transceiver.

3.2.2.2 Receivers - Normal receiver (*RCVR*) and Low-power receiver (*LPRCVR*)

RCVR and ***LPRCVR*** blocks are directly connected to CAN bus line. Thus, they have to be protected from high voltage hazard. ***RCVR*** is activated when the transceiver is in 'Normal mode', while ***LPRCVR*** is for 'Low-power mode'. The function is the same, but the propagation delay of ***LPRCVR*** is longer than ***RCVR***, and the output of ***LPRCVR*** is rather noisy because the current consumption is minimized in ***LPRCVR***.

RCVR block converts CANH/CANL signals and feed to RXD as shown in Fig. 3.8. Dominant bit is defined as logical low (0), and recessive bit is defined as logical high (1) in ISO 11898. Typical signal level of CANH and CANL is 3.5V and 1.5V, respectively, for dominant bit. For recessive bit, CANH and CANL both become 2.5V. Then, ***RCVR*** senses the difference between CANH and CANL, (CANH – CANL), and make a bit decision based on ISO 11898-2 definition.

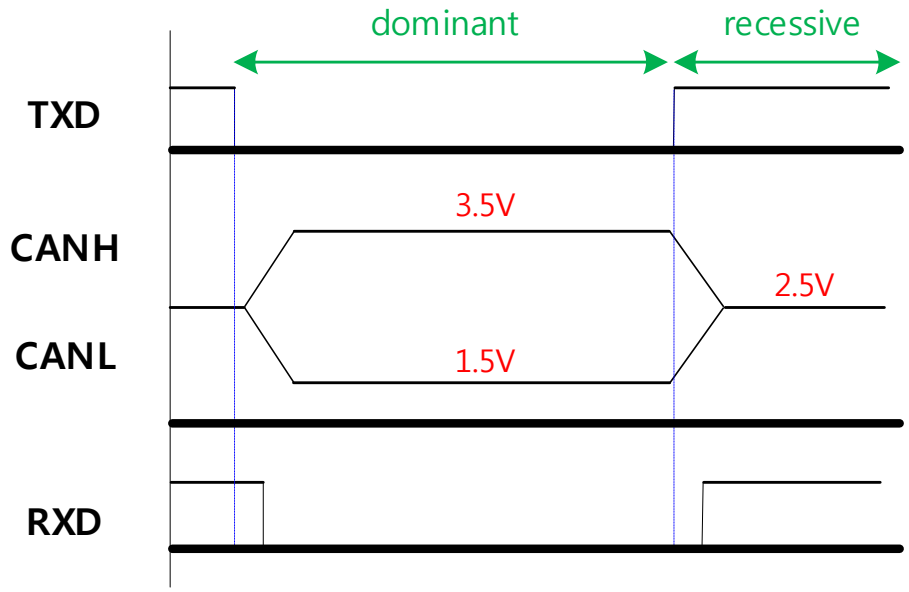


Figure 3.8 CAN bus levels

Fig. 3.9 shows sub-blocks in ***RCVR***. ***RxFEDiv*** is for high voltage protection. It is a just voltage divider scaling down the bus signal levels. ***RxFEOfs,*** a differential common source degeneration amplifier, is for making offset between *i_divh* and *i_divl*. It supports a current trimming feature to adjust the offset value.

$$\Delta V_{ofs} = 2 * 81.1k * I_{trim} \quad \text{(Eq. 3.1)}$$

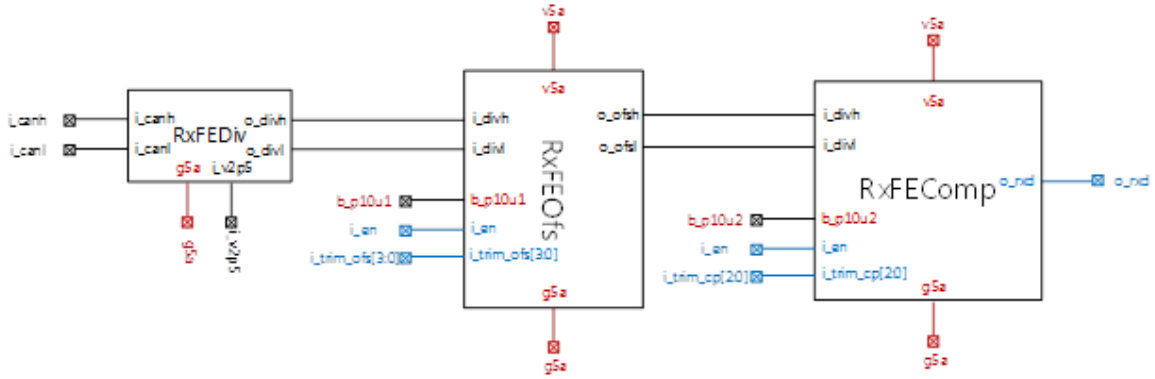


Figure 3.9 Block diagram of *RCVR*

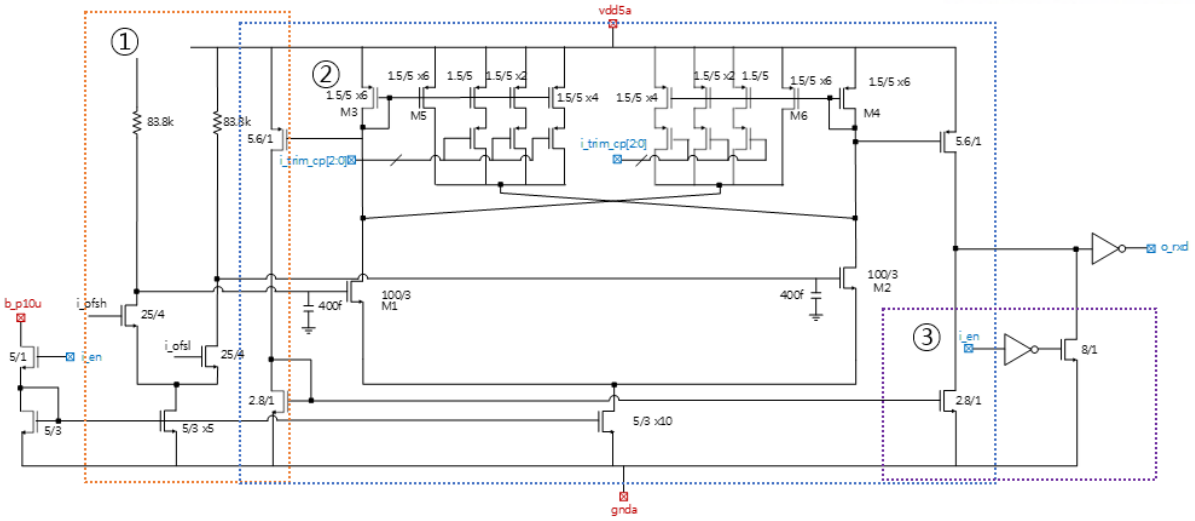


Figure 3.10 The schematic of *RxFEComp*

RxFEComp is a differential amplifier and a comparator with hysteresis. This is the main component of ***RCVR*** block. The box ① in Fig. 3.10 is the differential amplifier. It amplifies the input signal before making bit decision by hysteresis comparator of ②.

$$A_v = g_m * 83.8k = \sqrt{\frac{2\mu_n C_{ox} W}{L} I_D} * 83.8k = \sqrt{\frac{2\mu_n C_{ox} * 25}{4} * \frac{50u}{2}} * 83.8 = 6.54 \quad \text{(Eq. 3.2)}$$

Then, the common mode voltage is

$$V_{CM} = 5Va - 83.8k * 25u = 3V \quad \text{(Eq. 3.3)}$$

The hysteresis voltage of comparator is

$$V_{hys} = V_{TRP+} - V_{TRP-} = 70mV. \quad \text{(Eq. 3.4)}$$

The sub-block of ③ is a pull-down switch. The signal of the output inverter forces to be recessive bit when *i_en* is logical low.

Simulation condition:

- EN = logical high (1)
- 1Mbps data rate (pulse width 1us)
- Rising time and falling time of CANH/CANL = 100ns

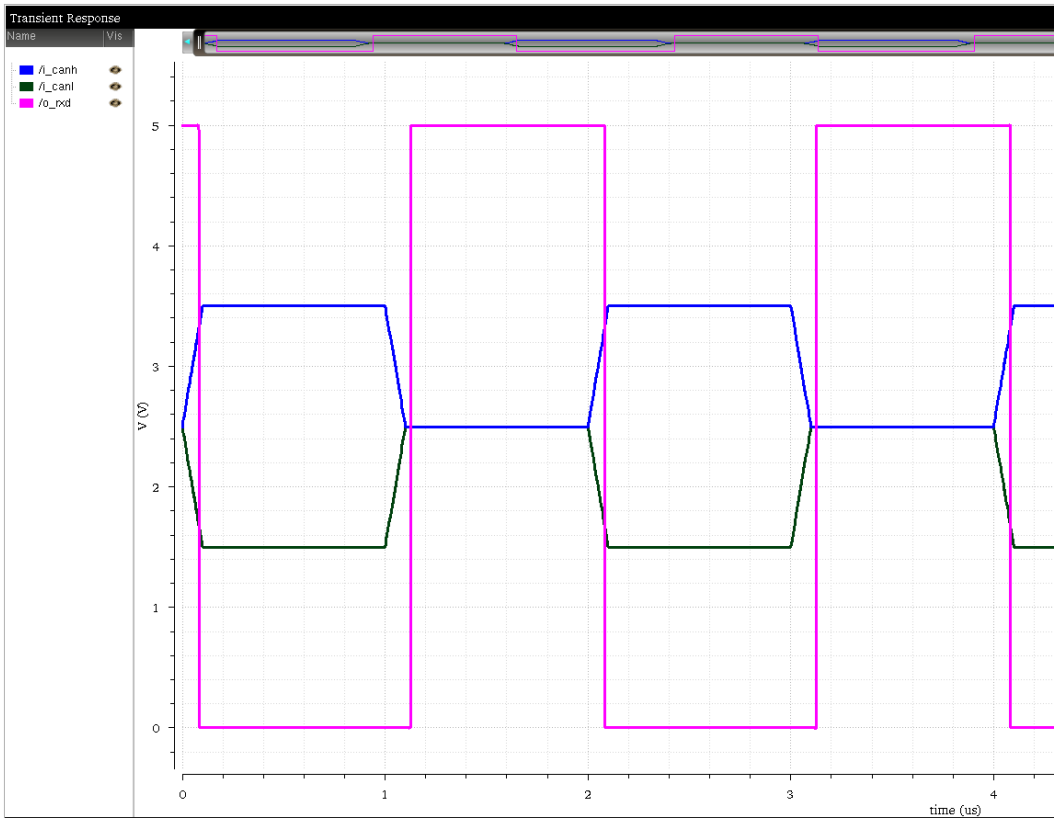
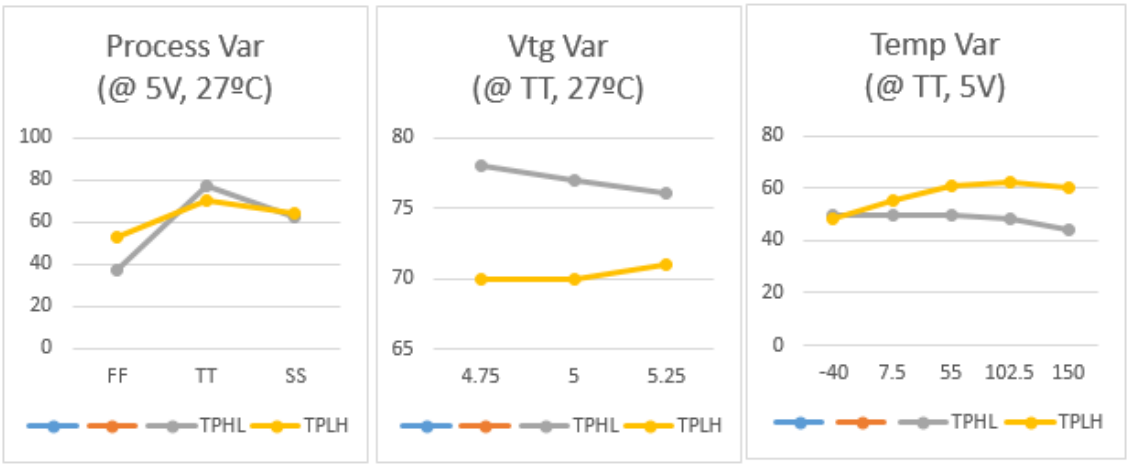


Figure 3.11 Simulation result of *RCVR* block



T_{PHL}: propagation delay high to low, TPLH: propagation delay low to high

Figure 3.12 Performance tendency of *RCVR* block in PVT simulation

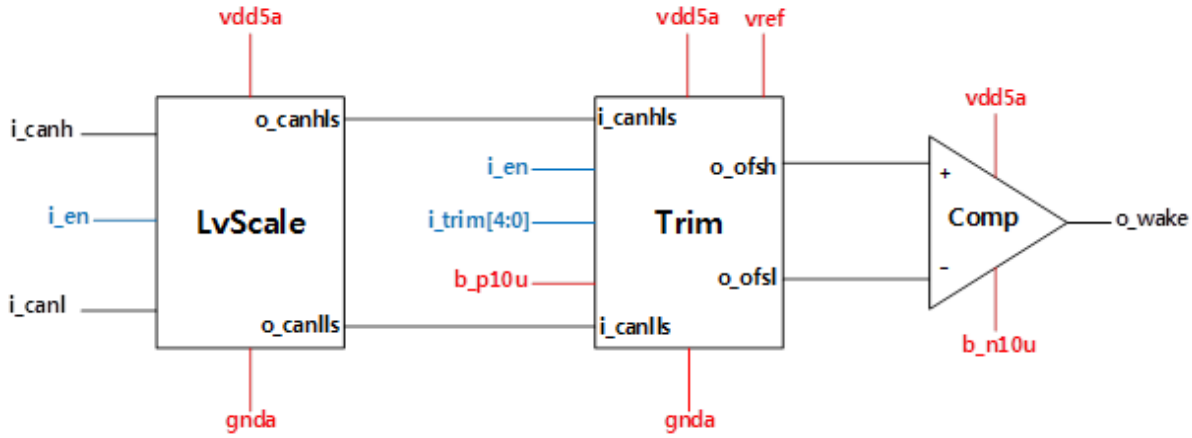


Figure 3.13 Block diagram of *LPRCVR*

LPRCVR is to meet ISO 11898-5, which specifies the remote wake-up feature. The transceiver goes to a standby mode to minimize the power consumption. For this, the normal receiver (***RCVR***) should be deactivated. However, the transceiver still needs a receiver due to the remote wake-up feature.

According to ISO 11898-5, the transceiver should wake up when the bus become busy. Hence, a separate receiver, ***LPRCVR,*** designed for low-power will be required, as shown in Fig. 3.13. There are 3 sub blocks. ***LvScale*** is similar with ***RxFEDiv*** of ***RCVR***, but ***LvScale*** don't use a resistor divider. ***Trim*** block is for current trimming and offset. ***Comp*** block is a comparator. The function is almost the same with ***RCVR***.

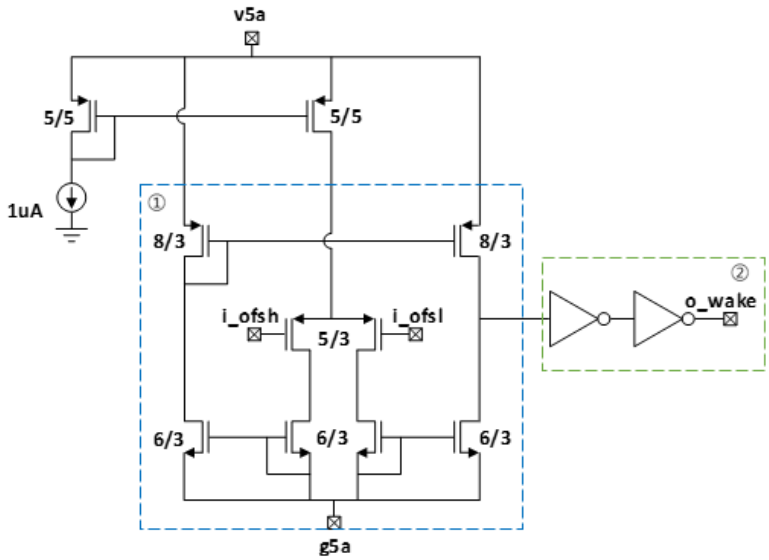


Figure 3.14 The schematic of *Comp*

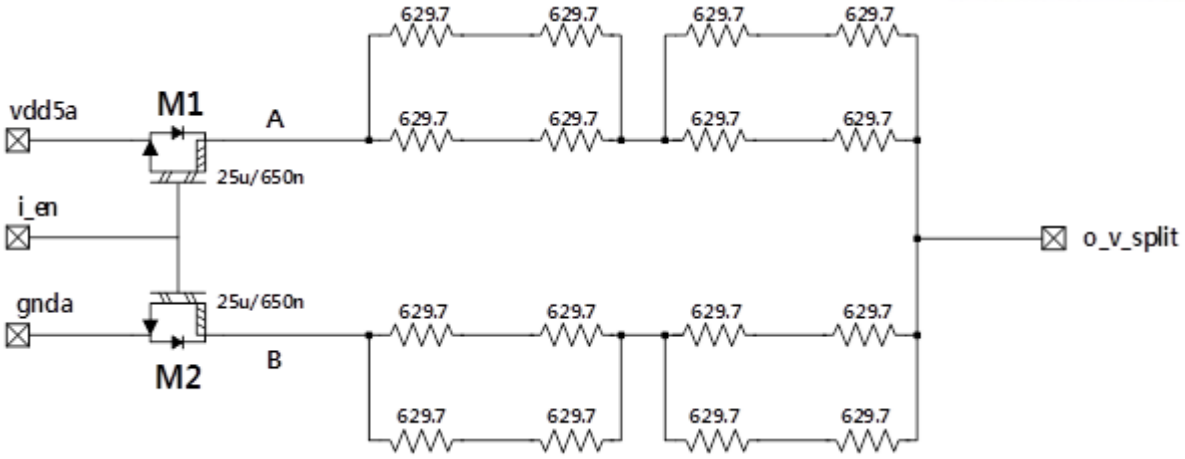


Figure 3.15 The schematic of *VSplit*

3.2.2.3 *VSplit* block

Fig. 3.15 shows the schematic of ***VSplit*** block. It is a voltage divider. It divides supply voltage (V_{DD5}) into half, therefore *o_v_split* would be 2.5V. The resistive value of high side is 1260Ω, and low side is the same. The value is 10 times larger than the bus termination resistors (typically 120Ω). Thus, it functions as weak voltage source for improving the EME performance.

M1 and M2 are switches for enabling the block. This MOS pair also has to be protected from high voltage hazard, because the output of *o_v_split* is connected to bus line. In this work, EDMOS pair was used in order to match the electrical characteristics with ***BI*** block (protection diodes and MOS).

The output current is as below.

$$\frac{1}{2}u_n C_{OX}\left(\frac{W}{L}\right)\{2(V_{GS}-V_{TH})V_{DS}-V_{DS2}\}=1.43mA \qquad \text{(Eq. 3.5)}$$

It could supply significantly less current than ***BI*** block, it weakly holds bus to a half of V_{DD} while bus is transmitting recessive bit.

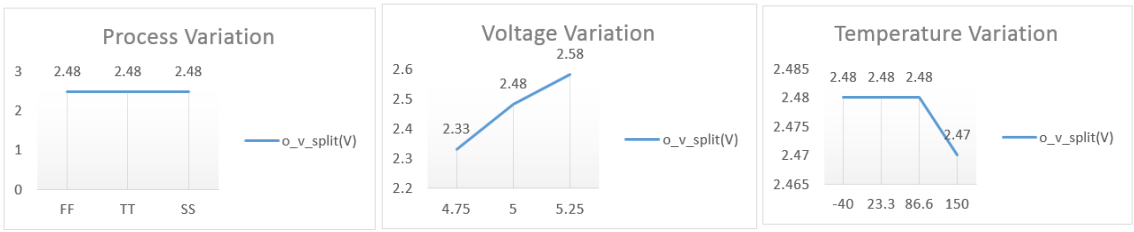


Figure 3.16 PVT variation simulation results (unloaded condition only)

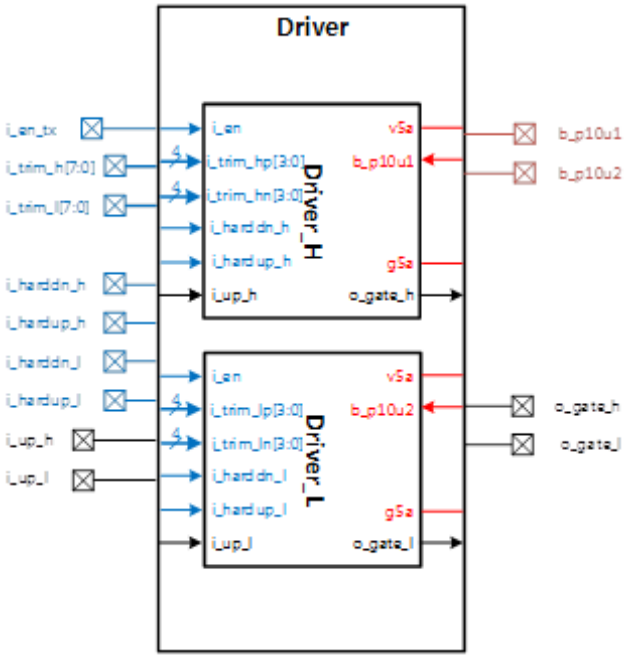


Figure 3.17 Block diagram of *Driver*

3.2.2.4 *Driver* block

Driver block is a push-pull gate driver for switch MOS in ***BI*** block. Since the size of switch MOS is quite large, and therefore the gate capacitance is as big as 10pF, ***Driver*** is a current driver, and supports 8-bit current trimming feature for slope control. ***Driver*** block is a pair of sub-driver, ***Driver_H*** and ***Driver_L*** as shown in Fig. 3.17. They were tuned by current trimming for high-side and low-side switch MOS respectively.

Fig. 3.18 shows the detailed circuit of ***Driver***. The sub-circuit of ① is current mirror for copying the bias current of 10uA came from ***Bias*** block. The sub-circuit ② is a push pull driver, and ③ is for current trimming. It supports 4-bit current trimming level for high side, and low side is same as it. The sub-circuit ④ is for special purpose.

When the driver is stopped suddenly during transmitting a dominant bit, then the charges is trapped in gate of the switch MOS. Therefore, the CAN bus is stuck at the dominant state for a long time, until it discharged naturally. Until that, any node couldn't communicate via the bus line. It is a critical problem which paralyzes the whole bus. Hence, ***Digital Logic*** block controls ***Driver*** through *i_hardup* and *i_harddn* pins, and it forces to leak the trapped charges.

There are some cases which cause this phenomenon; thermal shutdown, TXD dominant time-out, and sudden mode change from Normal to Stand-by mode.

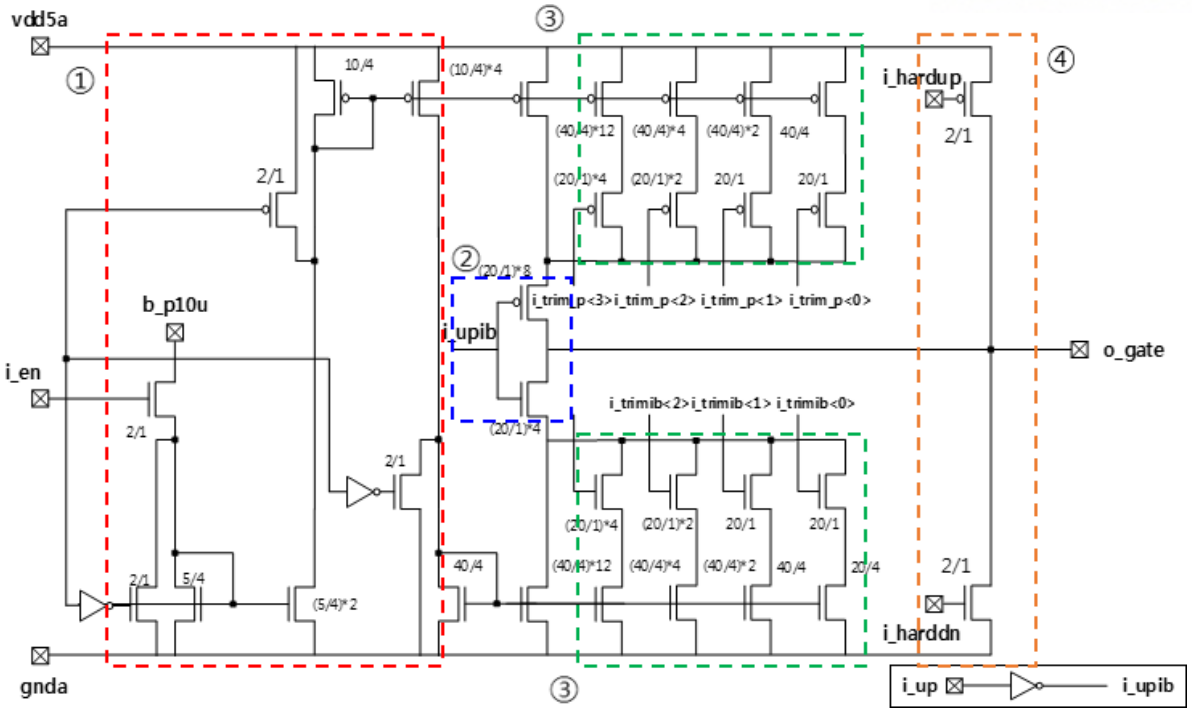


Figure 3.18 The schematic of *Driver_H* and *Driver_L*

The slope control is key feature at ***Driver*** block, and the rising time t_r could be derived as the following.

$$t_r = \left(\frac{CV}{I}\right) * 0.8 \ (because\ rise\ time\ calculated\ from\ 10\%\ to\ 90\%) \qquad \text{(Eq. 3.6)}$$

The rise time is commonly measured from 10% to 90%, so 80% is applied because it is linear system. The designed rising time is typically 40ns.

3.2.2.5 Thermal shutdown (*TSD*)

This block is for thermal shutdown. It prevents the device to be operating abnormally. When the temperature of IC is increased, the electrical characteristic is distorted. Some application is very sensitive to this temperature change. Automotive ICs designed to guarantee the performance with the temperature range from -40℃ to 160℃. Therefore, the IC should stop the operation when the temperature exceeds the guaranteed range. Refer to [9], [10], and [11] for more background about thermal protection. In this work, the triggering point was set to 165℃.

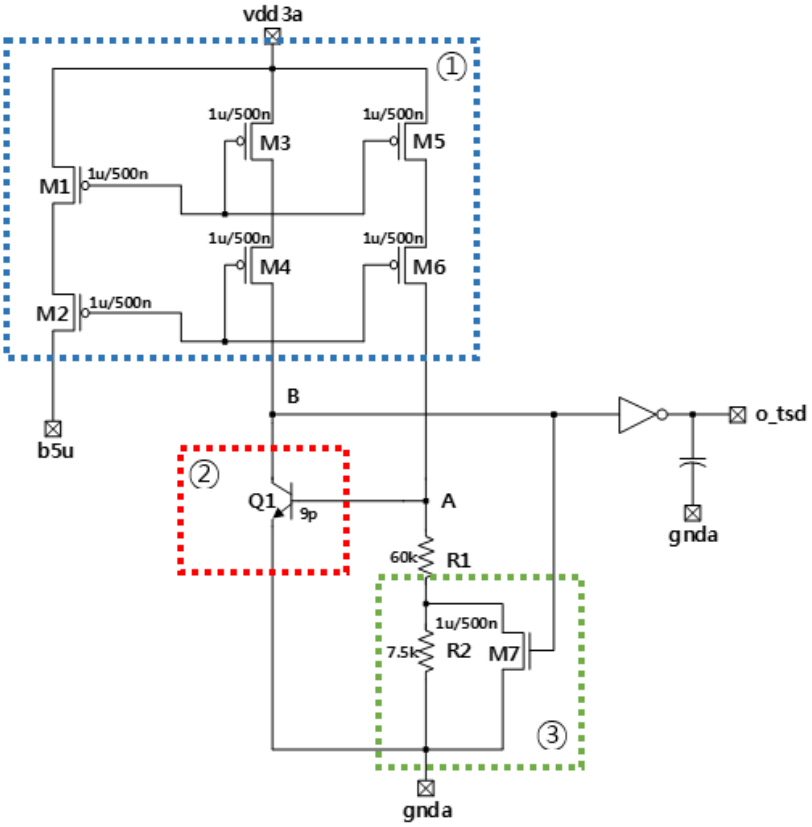


Figure 3.19 The schematic of *TSD*

In Fig. 3.19, the sub-circuit ① is current mirror which copy the bias current. The sub-circuit ② is temperature sensor. It is turned on and off by temperature change. The sub-circuit ③ is the hysteresis circuit.

At room temperature (typically 27℃), the voltage of node A, V_A, is smaller than $V_{BE,on}$. Thus, Q1 is turned off. Therefore, there is no current path on B node, so V_B is the same with V_{dd3a}. Thus, *o_tsd* would be 0V, i.e., logical low. At this time, M7 is on, so there is no voltage drop with R2.

As temperature goes up, $V_{BE,on}$ of Q1 would be reduced. However, the resistive value of R1 and R2 increases, so V_A would go up. As a result, Q1 is turned on, V_B should be around of 0V. Then, *o_tsd* would be a logical high.

$$V_A = I_{ref} \times R1 \times \{1 + \frac{\partial R}{\partial T} \times (T_{SD} - 0)\} \quad \text{(Eq. 3.7)}$$

$$V_{BE,on} = V_{BE,room} + \frac{\partial VBE}{\partial T} \times (T_{SD}\text{-}T_{room}) \quad \text{(Eq. 3.8)}$$

$\frac{\partial V_{BE}}{\partial T} = -1.85\text{mV/K}$, $\frac{\partial R}{\partial T_1} = 2.9m\Omega/K$, $\frac{\partial R}{\partial T_2} = 6u\Omega/K^2$, $V_{BE,room}$ = 561mV. The target T_{SD} is 165℃, therefore, R1 should be 56.2kΩ.

$$0.561 - 2.2m \times (165 - 0) = [R1 \times \{1 + 2.9m \times (165 - 27)\}]$$
$$\therefore R1 = 56{,}156\Omega \quad \text{(Eq. 3.8)}$$

For implementing the hysteresis of 10℃, change the target T_{SD} to 155℃. Add R2 at Eq. 3.7, and calculate it.

$$0.716 - 2.2m \times (155 - 27) = [(R1 + R2) \times \{1 + 2.9m \times (155 - 27)\}]$$
$$\therefore R2 = 4026.6\Omega \quad \text{(Eq. 3.9)}$$

Finally, tune the size for obtaining the target T_{SD}. Fig. 3.20 shows simulation result with hysteresis.

.

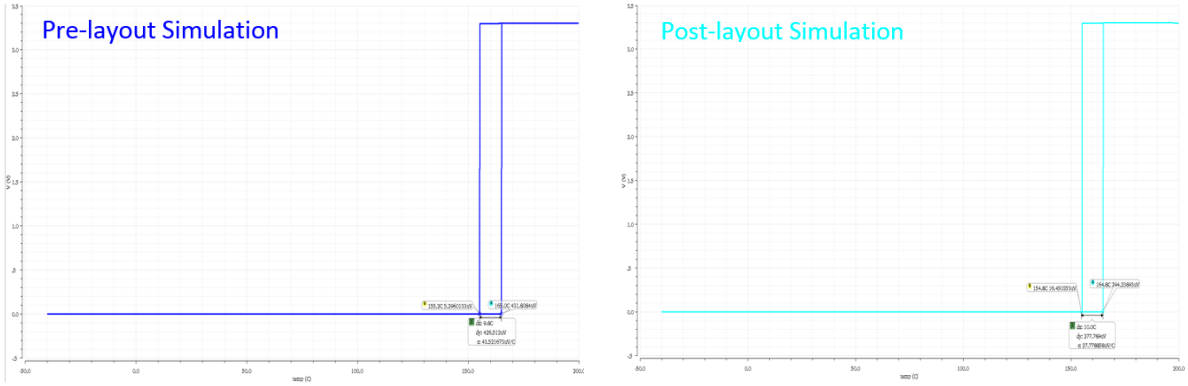
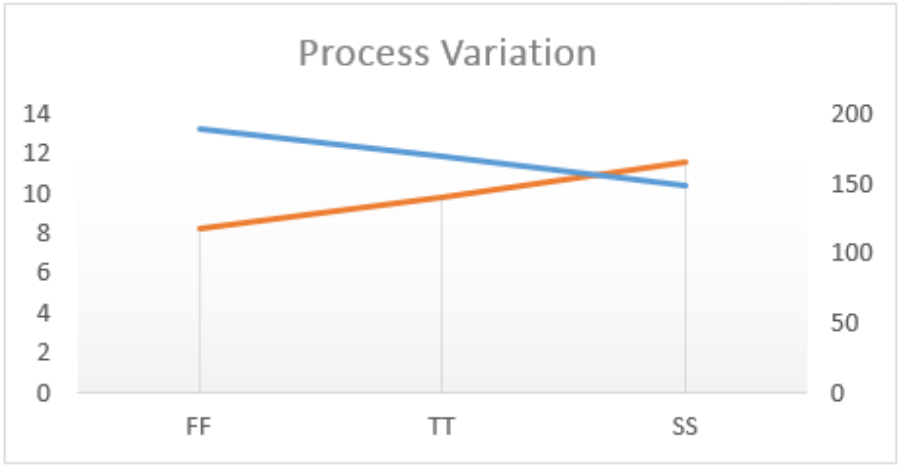
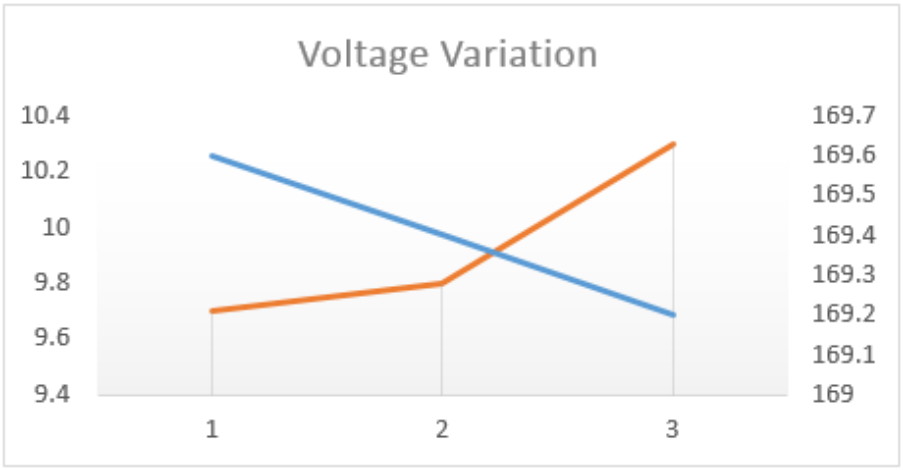


Figure 3.20 Simulation result (a) pre-layout simulation, (b) post-layout simulation





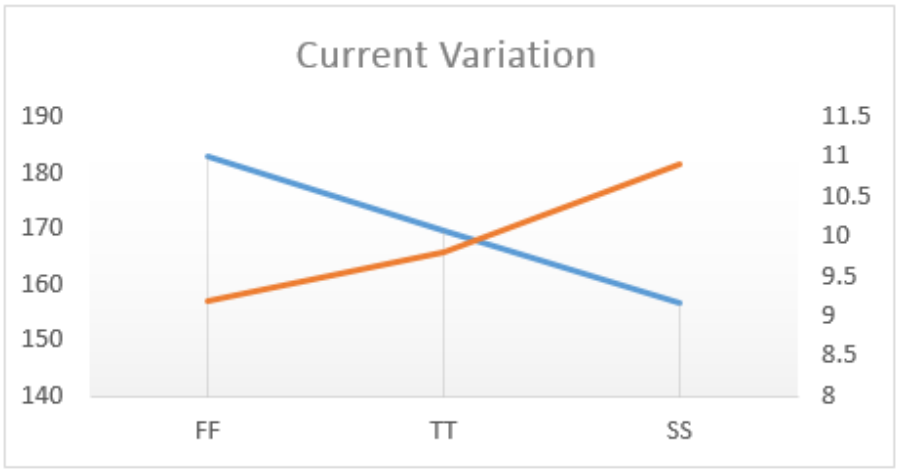


Figure 3.21 PVT simulation result: changing of trigger point (T_{SD})

3.2.2.6 Oscillator (*OSC*)

OSC block generates a clock signal of 4MHz (square wave with 50% duty ratio). It supports 4-bit current trimming control for tuning the frequency. The adjustable frequency range is from 3MHz to 5.8MHz. The generated clock is supplied to ***Digital Logic*** block.

Fig. 3.22 is a schematic of ***OSC***. I_0 is a bias current supply. M_1 and M_2 is for current mirror. First, the target frequency f_0 is 4MHz. Then, the system allows error range of 50%. The upper bound (f_{0H}) was set to 5.8MHz, and the lower bound (f_{0L}) was set to 3MHz. I_0 is 5uA.

$$M_0 : M_1 = 3 : 2 \qquad \text{(Eq. 3.10)}$$

$$I_0 : I_1 = 3 : 2, \qquad therefore\ I_0 = 3.33uA \qquad \text{(Eq. 3.11)}$$

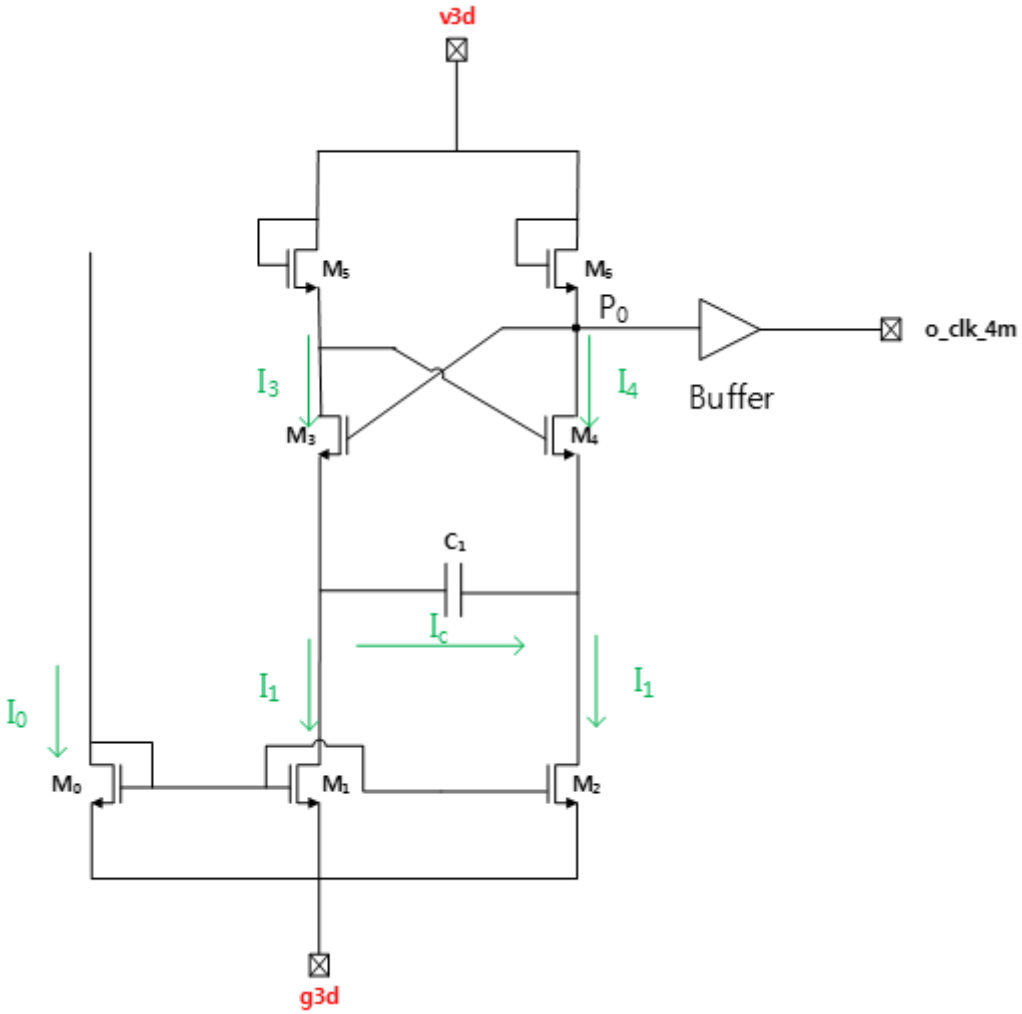


Figure 3.22 The schematic of *OSC* block

Table 3.4 Frequency tuning by current trimming (***OSC***)

Trimming code (i_trim[3:0])	**f_{osc} [MHz]**
0 0 0 0	3.000
0 0 0 1	3.000 + 0.187
0 0 1 0	3.000 + 0.187 * 2
…	…
0 1 0 1 (default)	3.000 + 0.187 * 5 = 3.935 ~ 4 (this value was confirmed by using simulation)
…	…
1 1 1 1	3.000 + 0.187 * 15 = 5.800

ΔV_C is obtained by simulation, the value is 0.75V.

$$f_{0L} = I_C / 4 C_1 \Delta V_C \quad \text{(Eq. 3.12)}$$

$$C_1 = I_C / 4 \Delta V_C \, f_{0L} = \frac{3.33\text{uA}}{(4 \times 3 \times 10^6 \times 0.75)} = 370 fF. \quad \text{(Eq. 3.13)}$$

This is the value of C_1, but it might be tuned slightly by simulation. It supports a 4-bit current trimming, and the unit step of current trimming is 0.187MHz. It is calculated by Eq. 3.14.

$$f_{unit} = \frac{f_{oH} - f_{oL}}{2^4 - 1} = 0.187\text{MHz}. \quad \text{(Eq. 3.14)}$$

Refer to Table 3.4 for the detailed frequency control by trimming bit code. Fig. 3.23 describes a simulation condition for ***OSC***. The rising time of supply voltage (t_r) is set to 1ms.

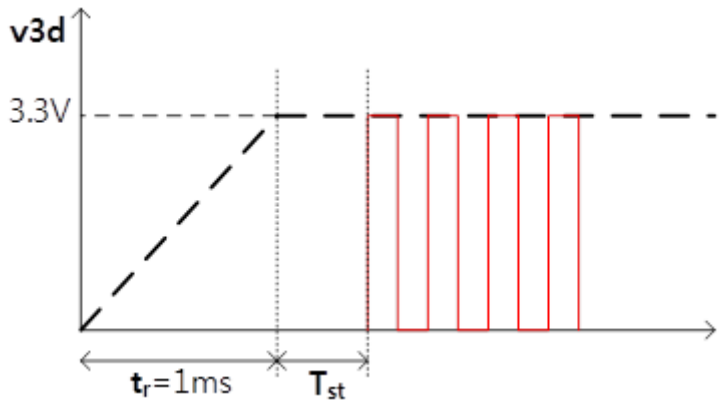
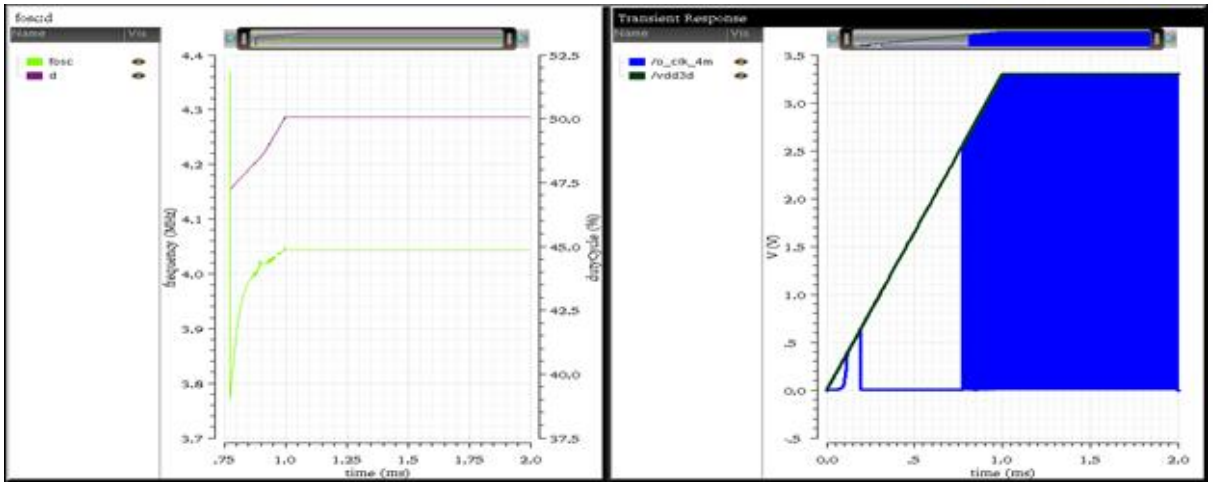
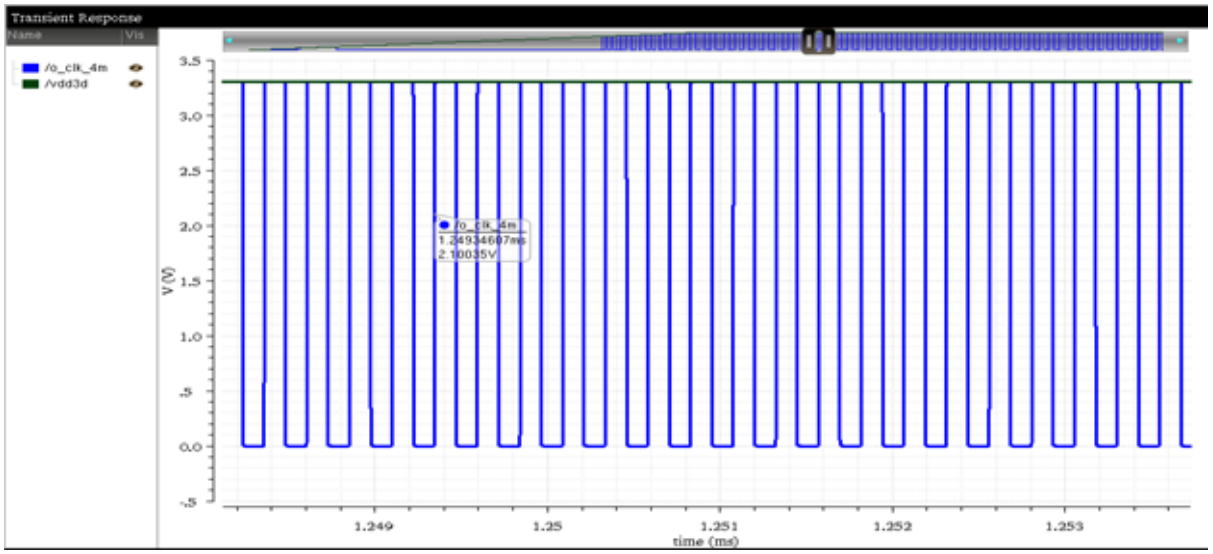


Figure 3.23 Output clock would be generated after supply voltage (v3d) is stabilized



(a) (b)



(c)

Figure 3.24 Simulation results (a) duty ratio (50%), (b) output clock when powered on, (c) clock pulse period

Table 3.5 PVT simulation result (***OSC***)

1. Process Variation						
Process	Voltage (v3d)	Temperature	f_{osc}	d	I_{cc}	T_{st}
TT	3.3V	27℃	4.06MHz	50.00%	35uA	0ns
FF			5.13MHz	50.90%	51.8uA	0ns
SS			3.28MHz	49.00%	28.2uA	0ns
2. Voltage Variation						
Process	Voltage (v3d)	Temperature	f_{osc}	d	I_{cc}	T_{st}
TT	3.135V	27℃	4.04MHz	49.20%	33uA	0ns
	3.3V		4.06MHz	50.00%	35uA	0ns
	3.465V		4.07MHz	50.70%	37.8uA	0ns
3. Temperature Variation						
Process	Voltage (v3d)	Temperature	f_{osc}	d	I_{cc}	T_{st}
TT	3.3V	-40℃	4.67MHz	51.60%	37.6uA	0ns
		27℃	4.06MHz	50.00%	35uA	0ns
		150℃	3.71MHz	46.90%	34uA	0ns
Highest Frequency						
Process	Voltage (v3d)	Temperature	f_{osc}	d	I_{cc}	T_{st}
FF	3.465V	-40℃	5.84MHz	53.70%	60.3uA	0ns
Lowest Frequency						
Process	Voltage (v3d)	Temperature	f_{osc}	d	I_{cc}	T_{st}
SS	3.135V	150℃	2.83MHz	46.40%	26.1uA	0ns

3.2.2.7 Level Shifter (*LvShift*)

LvShift converts the voltage level of signals. As mentioned in Section 3.2.1, there are three voltage domains; 1.8V, 3.3V, and 5V. Hence, the level shifter is required for moving into another voltage domain. The number of cases is 6.

- 1.8V to 3.3V
- 1.8V to 5V
- 3.3V to 5V
- 3.3V to 1.8V
- 5V to 3.3V
- 5V to 1.8V

Fig. 3.25 shows the level-up shifter with a buffer stage omitted. When the input voltage IN is logical low, MN1 is off and MN2 is on. The gate voltage of MP1 would be pulled down to ground. Then, MP1 will be on, and MP2 will be off since the gate voltage of MP2 is V_{DDH}. Therefore, the node A is pulled down to ground. In this case, the output will be low.

Then, IN is logical high, MN1 is on and MN2 is off. Then, the gate voltage of MP2 would be pulled down to ground, and MP2 would be on. Thus, the node A will be logical high. Table 3.6 summarizes this operation.

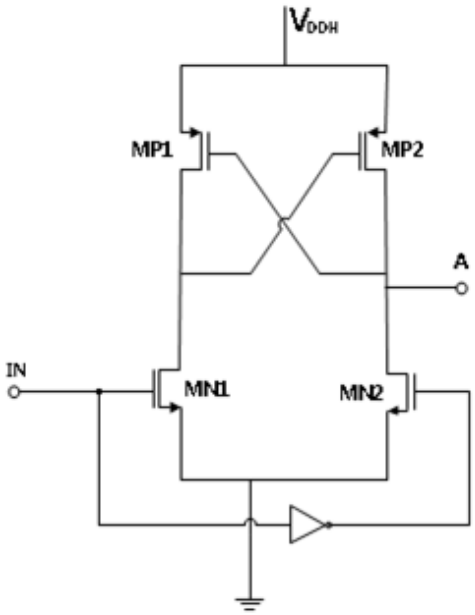


Figure 3.25 The basic schematic of level-up shifter

Table 3.6 Operation summary

IN	MN1	MN2	MP1	MP2	A (output)
Logical low	OFF	ON	ON	OFF	LOW(GND)
Logical high	ON	OFF	OFF	ON	HIGH(V_{DDH})

For the optimized level shifter, the propagation delay should be minimized to reduce path delay from TXD to RXD. The high-to-low propagation delay could be modeled as Eq. 3.15. The delay between the differential inputs was approximated as zero.

$$t_{PHL} = \frac{\frac{C_L V_{DDH}}{2}}{\frac{1}{2} u_n C_{OX} \frac{W_n}{L_n} (V_{DD} - V_{TH,n})^2} = \frac{C_L V_{DDH}}{u_n C_{OX} \frac{W_n}{L_n} (V_{DD} - V_{TH,n})^2} \quad \text{(Eq. 3.15)}$$

Similarly, the low-to-high propagation delay could be calculated by Eq. 3.16.

$$t_{PLH} = \frac{\frac{C_L V_{DDH}}{2}}{\frac{1}{2} u_p C_{OX} \frac{W_p}{L_p} (V_{DD} - V_{TH,p})^2} = \frac{C_L V_{DDH}}{u_p C_{OX} \frac{W_p}{L_p} (V_{DD} - V_{TH,p})^2} \quad \text{(Eq. 3.16)}$$

Since the total delay is minimized, total propagation delay could be defined as Eq. 3.17. Then, find the lowest value by adjusting the width and length of MN1, MN2, MP1, and MP2. The length will be fixed to minimum value (1um) for minimizing the size of a level shifter.

$$t_P = \frac{t_{PHL} + t_{PLH}}{2} = \frac{C_L V_{DDH}}{u_n C_{OX} \frac{W_n}{L_n} (V_{DD} - V_{TH,n})^2} + \frac{C_L V_{DDH}}{u_p C_{OX} \frac{W_p}{L_p} (V_{DD} - V_{TH,p})^2} \quad \text{(Eq. 3.17)}$$

Table 3.7 Electrical characteristic of device (27℃)

Device	C_{OX}	$\mu_n(\mu_p)$	V_{th}
xnch_tk33	3.61 fF/um^2	194.2cm^2/Vs	0.530V
xpch_tk33	3.42 fF/um2	94.3cm^2/Vs	-0.666V
xnch_svn	1 fF/um^2	422.4cm^2/Vs	0.812V
xpch_svp	1 fF/um^2	156.7cm^2/Vs	-0.780V

C_L is the load capacitance of buffer stage. W' and L' are width and length of the buffer stage.

$$C_L = C_{OX}\left(W'_p L_p\right) + C_{OX}\left(W'_n L_n\right) = \left(\frac{1fF}{um^2}\right)(1.4um^2 + 2.8um^2) = 4.2fF. \quad \text{(Eq. 3.18)}$$

From Eq. 3.17 and Eq. 3.18, the propagation delay t_p is

$$t_P = 0.0509 \times \frac{W'_p + W'_n}{W_n} + 0.0342 \times \frac{W'_p + W'_n}{W_p} \text{ (ns)} \quad \text{(Eq. 3.19)}$$

The mobility ratio of PMOS and NMOS is commonly approximated as 1:3, so W'_p:W'_n should be 3:1. $W`'_p$ and $W`_n$ should be minimized for reducing load capacitance of buffer stage. Hence $W`'_p$ = 7.5um, and $W`_n$ = 2.5um.

$$t_P = 0.0509 \times \frac{10um}{W_n} + 0.0342 \times \frac{10um}{W_p} \text{ (ns)} \quad \text{(Eq. 3.20)}$$

The same value of T_{PHL} and T_{PLH} is desirable. For this, the saturation current should be same.

$$\frac{1}{2}u_n C_{OX}\frac{W_n}{L_n}(V_{DD} - V_{TH,n})^2 = \frac{1}{2}u_p C_{OX}\frac{W_p}{L_p}(V_{DD} - V_{TH,p})^2 \quad \text{(Eq. 3.21)}$$

The length was set to minimum value of 1um, therefore the relation between W_n and W_p is

$$\frac{W_n}{W_p} = 1.488 \approx 1.5 \quad \text{(Eq. 3.22)}$$

By using Eq. 3.23,

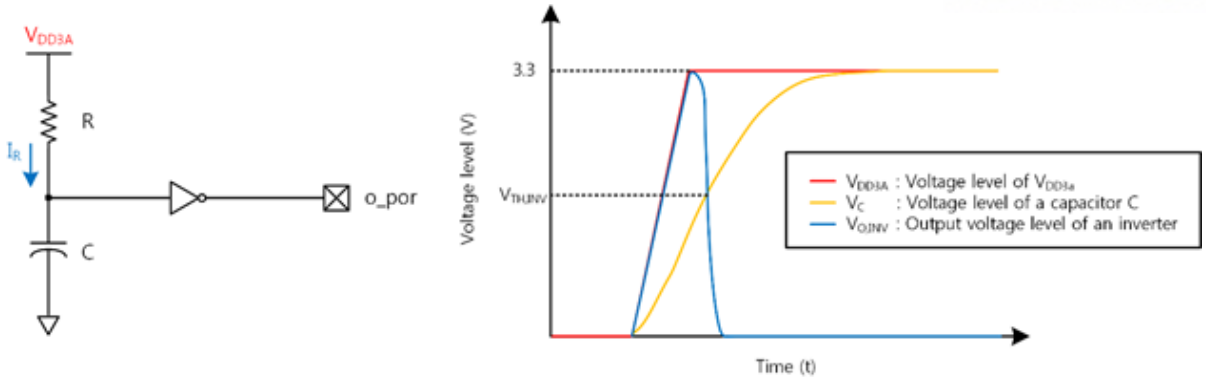
$$t_P = 0.0509 \times \frac{10um}{W_n} + 0.0342 \times \frac{15um}{W_n} = \frac{1.022}{W_n}\text{(ns)} \quad \text{(Eq. 3.23)}$$

For minimizing the size of level shifter, W_n was fixed to 9um, and W_p was fixed to 6um. As a result, t_P was 0.1136ns.

3.2.2.8 Power-on Reset (*PoR*)

PoR provides a reset pulse when the device is powered up. The reset signal initializes the memory elements in the transceiver.

The principle of ***PoR*** is RC time constant. It includes a resistor which is limiting the current, and a capacitor which is making a delay, and an inverter which is converting the analog level to digital. Refer to Fig. 3.26.



(a) (b)

Figure 3.26 (a) A basic PoR circuit, (b) Timing diagram of the PoR circuit

The basic operation is as follows:

(1) The circuit is in unpowered state (assume enough time was elapsed to discharge C completely)

(2) Power is on (V_{DD3A} = 3.3V)

(3) The capacitor C starts to charge.

(4) The output of the inverter is logical high during initial charging state.

(5) The output of the inverter becomes logical low when the voltage level of capacitor C is enough to trigger the inverter. ($V_C > V_{TH,INV}$, V_C: the voltage of the capacitor, $V_{TH,INV}$: the threshold voltage of the inverter)

(6) The capacitor C is fully charged.

(7) End of process.

The rising time of supply voltage is one of design considerations. Because the rising time is quite long, the capacitor C would be fully charged already before making a reset pulse. Table 3.8 shows the expected rising time of V_{DD} in the industry. This is for 5V applications.

Table 3.8 Estimated rising time of supply voltage in electronics industry

Reference sources	Supply voltage (V_{DD})	Estimated rising time
[13]	5V	Less than 50ms
[14]		Less than 100ms
[15]		Less than 20us

Table 3.9 Examples of power-on reset duration

Reference sources	PoR type	Reset pulse width	Notes
[15]	External PoR	Longer than 140ms	
[14]	Integrated on-chip PoR	256us	operating freq. is 4MHz
[16]	Integrated on-chip PoR	Sub-microseconds	For low-voltage applications

The second design consideration is duration of the reset pulse. Table 3.9 shows some examples which is used in industry. According to [14], they suggest to count the clock pulses, 1024 times for a reset pulse width. Because the internal oscillator needs some time to be stabilized. The proposed CAN transceiver's operating frequency is 4MHz. Hence, the required reset duration is 256us.

$$250ns \times 1024\ pulses = 256us. \quad \text{(Eq. 3.24)}$$

The target specification was set as below.

- The rising time of supply voltage: less than 100ms
- The duration of the reset pulse: longer than 256us

The basic structure of PoR introduced in Fig. 3.26 includes a resistor R. In IC design, R is not efficient for current limiting. Therefore, a modified structure is depicted in Fig. 3.27. The resistor R is replaced with PMOS M_0.

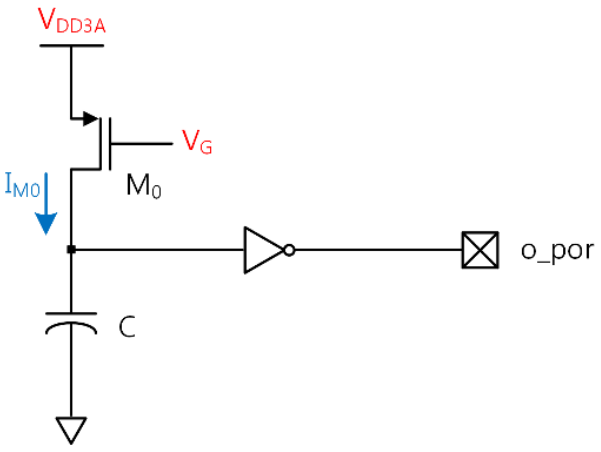


Figure 3.27 Modified PoR structure

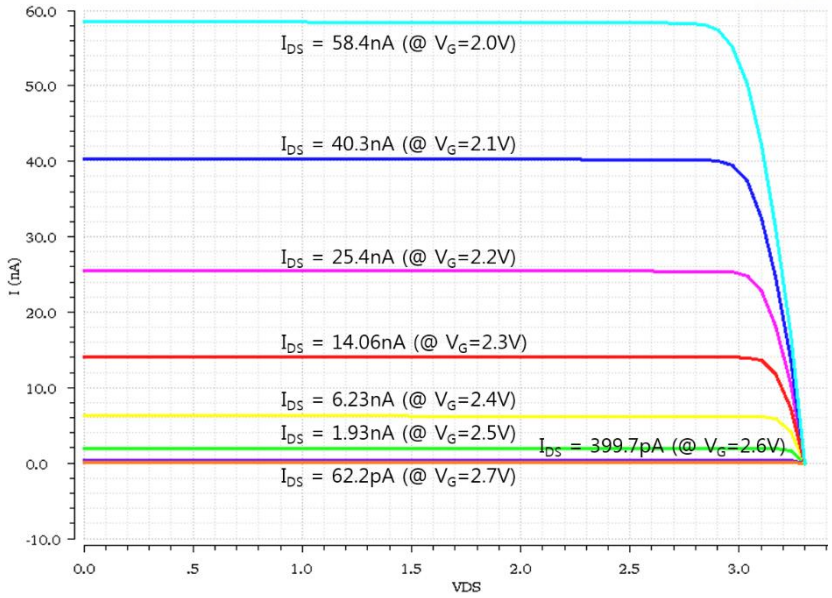


Figure 3.28 I_{DS} vs V_{DS} graph of M_0 (device: xpch_tk33)

The current I_{M0} should be minimized. Because as I_{M0} is reduced, the required capacitor size for achieving same reset pulse duration is also reduced. Therefore, the smallest current is desirable for small chip. Then, the minimum current of M_0 should be checked. The mobility of PMOS is smaller than NMOS, so PMOS device (xpch_tk33) was selected for designing.

Fig. 3.28 shows I_{DS} curve of M_0 (with 20um length, 400um width). The minimum current is checked as 62.2pA when V_G is 2.7V. However, it is almost tuned off state, thus V_G of 2.5V was selected for bias point. Fig. 3.28 was extracted only with the specified length of 20um. Fig. 3.29 shows the simulation results to obtain an optimized length

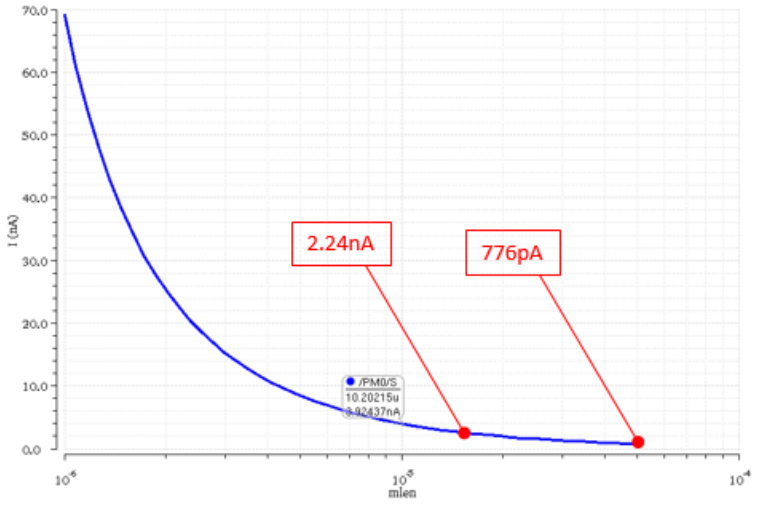


Figure 3.29 I_{DS} vs length of M_0 (from 1um to 50um)

The length of 10um is a good trade-off point for area efficiency, but the current might be a problem. The capacitor C has to be larger if the current is increased for achieving the same RC time constant.

To consider the size of M0 and C together,

$$t = RC = \frac{V}{I}C = \frac{CV}{I}. \quad \text{(Eq. 3.25)}$$

$$t = \frac{CV_{DD3A}}{IM0} = \frac{\left(C_{OX,C} \times W_C \times L_C\right) \times V_{DD3A}}{\frac{1}{2}\mu_P C_{OX,P} \frac{W_P}{L_P}(|V_{GS}| - |V_{TH,P}|)^2}. \quad \text{(Eq. 3.26)}$$

$$t = \frac{CV_{DD3A}}{IM0} = K\frac{W_C L_C}{\frac{1}{L_P}} = KW_C L_C L_P, \; where \; K = \frac{C_{OX,C} V_{DD3A}}{\frac{1}{2}\mu_P C_{OX,P} W_P (|V_{GS}| - |V_{TH,P}|)^2}. \quad \text{(Eq. 3.27)}$$

If the length (L_p) of M_0 is reduced by a half, the area ($W_C L_C$) of C is doubled for a same RC time constant.

$$A_{TOTAL} = A_C + A_P = W_c L_C + W_P L_P = \frac{t}{KL_P} + W_P L_P. \quad \text{(Eq. 3.28)}$$

The values of electrical parameters are described in Table 3.10.

$$A_{TOTAL} = \frac{t}{KL_P} + W_P L_P = \frac{1.045 \times 10^{-16}}{L_P} + 0.4 \times 10^{-6} \times L_P. \quad \text{(Eq. 3.29)}$$

Table 3.10 Electrical parameters for design

Symbol	Value	Unit
t	1	ms
$C_{OX,C}$ ($C_{OX,N}$)	1.97E-03	F/m^2
V_{DD3A}	3.3	V
u_P	7.12E-04	m^2/Vs
$C_{OX,P}$	2.07E-03	F/m^2
W_P	400	nm
V_G	2.5	V
$\|V_{GS}\|$	0.8	V
$\|V_{TH,P}\|$	0.752	V

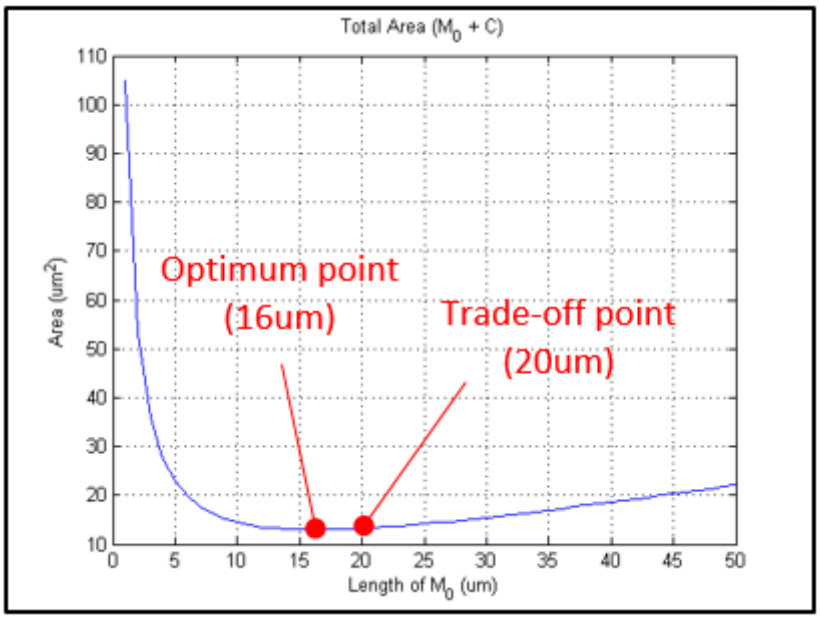


Figure 3.30 Total area of M_0 and C

As shown in Fig. 3.30, The length of 16um is the optimum point. and corresponding current is 2.24nA. the length of 20um was selected for the process reliability instead

To generate the bias voltage for V_G, the series of diodes was used, as shown in Fig. 3.31. The current through diodes could be reduced by adjusting the length of diodes. Also, by extending the length of diode D_4, V_G could be 2.5V.

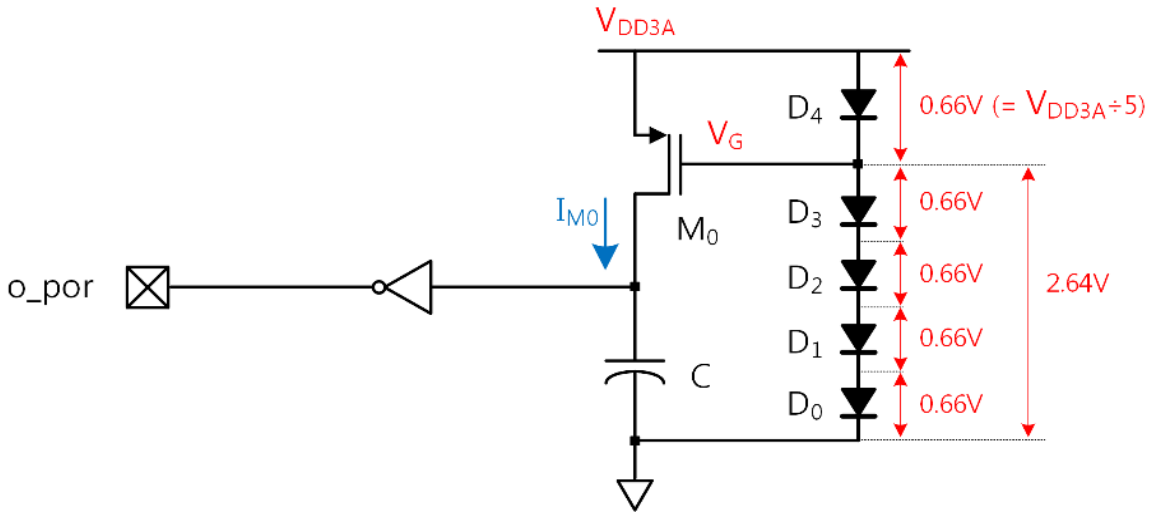


Figure 3.31 Basic concept for generating bias voltage

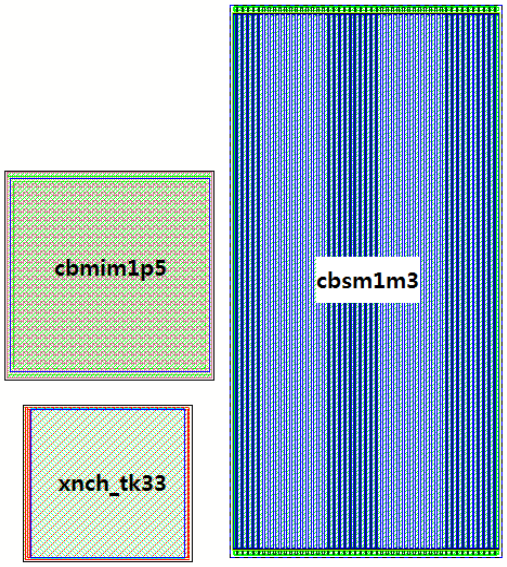


Figure 3.32 The area of device comparison for 1pF

The capacitance of C is determined by the charging speed and threshold value of inverter. The inverter could be replaced by Schmitt trigger. It is easy to adjust the threshold voltage. The higher threshold voltage induces more delay. Thus the required capacitor size is decreased. Then, the capacitor C could be replaced with MOS capacitor, and the diodes are also replaced by MOSFETs. The result is depicted in Fig. 3.33.

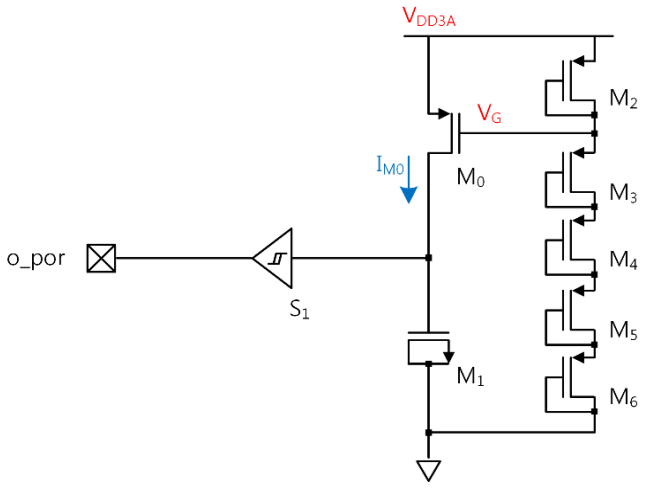
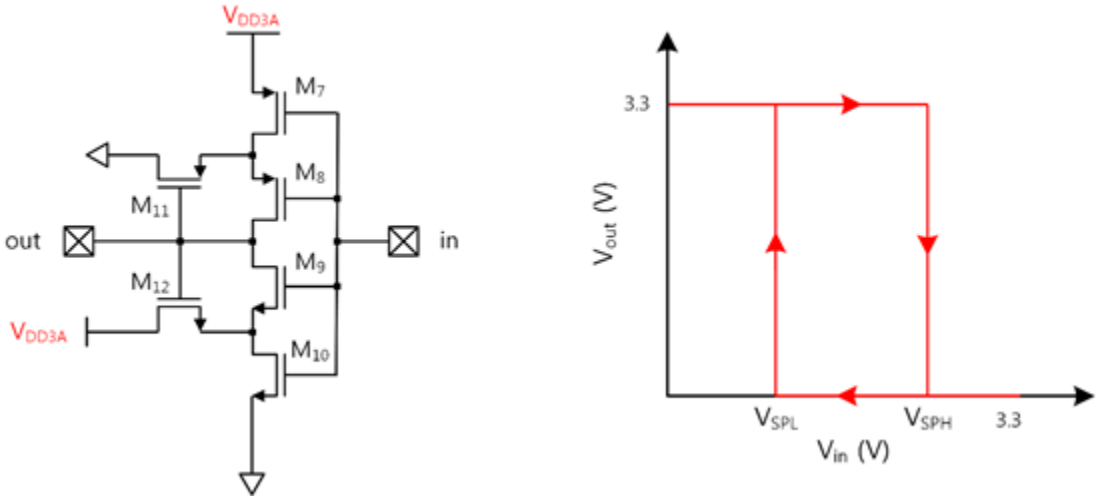


Figure 3.33 The inverter is replaced with Schmitt trigger and the capacitor is realized by a MOS capacitor. The diodes are replaced with diode-connected MOSFETs.



(a) **(b)**

Figure 3.34 (a) The circuit of a Schmitt trigger [22], (b) Behavior of a Schmitt trigger

The basic structure of a Schmitt trigger is depicted in Fig. 3.34. The reset pulse duration depends on RC time constant and V_{SPH} (switching point to high level) of the Schmitt trigger. V_{SPH} could be calculated by Eq. 3.30.

$$\frac{k_{10}}{k_{12}} = \left(\frac{V_{DD3A} - V_{SPH}}{V_{SPH} - V_{TN}}\right)^2, \quad where\ V_{TN}\ is\ a\ threshold\ voltage\ of\ NMOS,\ and\ k_x\ is\ a\ transconductance\ parameter. \tag{Eq. 3.30}$$

The next consideration is the required area for the Schmitt trigger and the MOS capacitor. The equation of the required area could be derived by modifying Eq. 3.30.

$$\sqrt{\frac{k_{10}}{k_{12}}} = \frac{V_{DD3A} - V_{SPH}}{V_{SPH} - V_{TN}} \tag{Eq. 3.31}$$

$$\sqrt{\frac{k_{10}}{k_{12}}}(V_{SPH} - V_{TN}) = V_{DD3A} - V_{SPH} \tag{Eq. 3.32}$$

$$\left(\sqrt{\frac{k_{10}}{k_{12}}} + 1\right) V_{SPH} = V_{DD3A} + \sqrt{\frac{k_{10}}{k_{12}}} V_{TN} \tag{Eq. 3.33}$$

$$V_{SPH} = \frac{V_{DD3A} + \sqrt{\frac{W_{10}/L_{10}}{W_{12}/L_{12}}}V_{TN}}{\left(\sqrt{\frac{W_{10}/L_{10}}{W_{12}/L_{12}}} + 1\right)}. \quad \text{(Eq. 3.34)}$$

L_{10} and L_{12} is fixed as the same value of 350nm for minimizing the size. Then,

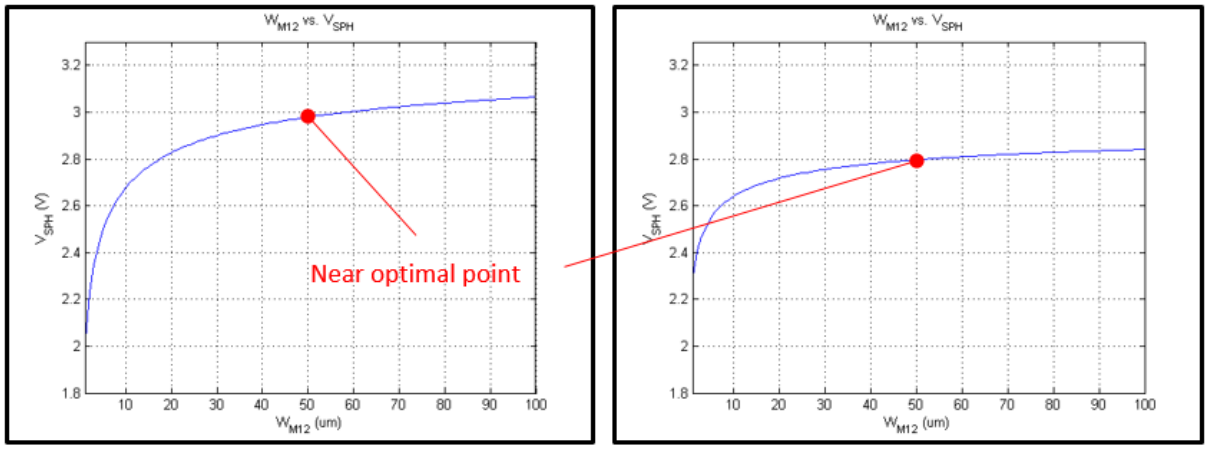
$$V_{SPH} = \frac{V_{DD3A} + \sqrt{W_{10}/W_{12}}\,V_{TN}}{\left(\sqrt{W_{10}/W_{12}} + 1\right)}. \quad \text{(Eq. 3.35)}$$

Eq. 3.35 determines V_{SPH}. W_{10} is fixed to the small and reliable value of 1um, so now the variable is only W_{12}. However, the simulated value of V_{SPH} was different with Eq. 3.35. Eq. 3.35 is modified for compensating the mismatch.

$$V'_{SPH} = \alpha\left(\frac{V_{DD3A}+\sqrt{W_{10}/W_{12}}V_{TN}}{(\sqrt{W_{10}/W_{12}}+1)}\right) + \beta, \ \ where\ \alpha = 0.52,\ \beta = 1.2474. \quad \text{(Eq. 3.36)}$$

The area of a Schmitt trigger A_S is determined by just W_{10} and W_{12}. Because W_{10} and W_{12} are only variable, the other is fixed constant. Therefore, the area of a Schmitt trigger could be approximated by Eq. 3.37.

$$A_S = W_{10}L_{10} + W_{12}L_{12}. \quad \text{(Eq. 3.37)}$$



(a) (b)

Figure 3.35 (a) W_{12} vs V_{SPH} (before compensation), (b) W_{12} vs V'_{SPH} (after compensation)

Less capacitance would be needed if higher V_{SPH} was selected. However, the high V_{SPH} induces larger size of Schmitt trigger. To find the optimum point of V_{SPH}, modify Eq. 3.25.

$$t = \frac{CV}{I} = \frac{CV_{SPH}}{I_{M0}}. \quad \text{(Eq. 3.38)}$$

$$C = \frac{I_{M0}t}{V_{SPH}} = C_{OX,N}W_C L_C. \quad \text{(Eq. 3.39)}$$

Then, the area of the MOS capacitor A_C is determined as below.

$$A_C = W_c L_C = W_1 L_1. \quad \text{(Eq. 3.40)}$$

Thus, to total area of the Schmitt trigger and the MOS capacitor is

$$\begin{aligned} A_{TOTAL} &= A_S + A_C = W_{10}L_{10} + W_{12}L_{12} + W_1 L_1 \\ &= (1um \times 350nm) + W_{12}(350nm) + W_1 L_1 \end{aligned} \quad \text{(Eq. 3.41)}$$

By modifying Eq. 3.30,

$$\frac{k_{10}}{k_{12}} = \frac{W_{10}}{W_{12}} = \frac{1um}{W_{12}} = \left(\frac{V_{DD3A} - V_{SPH}}{V_{SPH} - V_{TN}}\right)^2 \quad \text{(Eq. 3.42)}$$

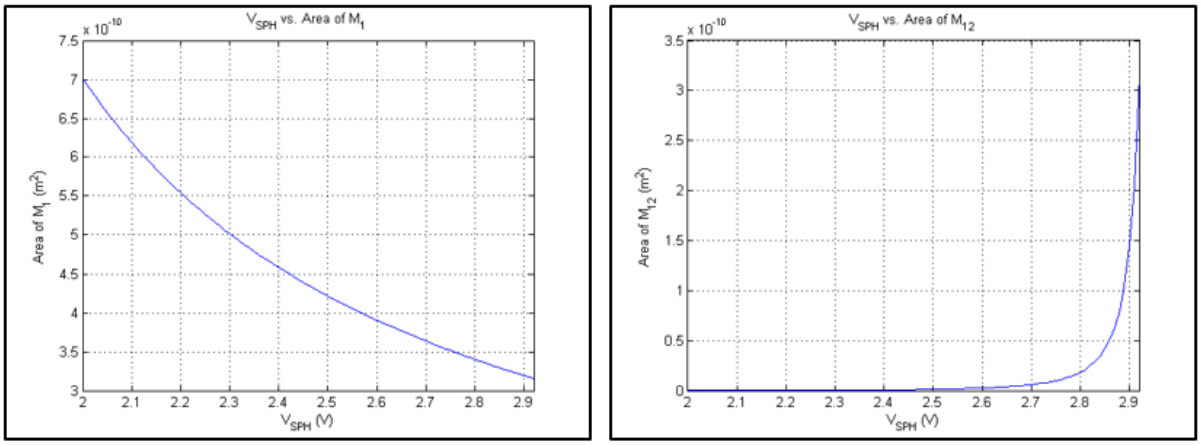
$$W_{12} = \frac{1um}{\left(\frac{V_{DD3A} - V_{SPH}}{V_{SPH} - V_{TN}}\right)^2}. \quad \text{(Eq. 3.43)}$$

Then, Eq. 3.40 can be transformed as below.

$$W_1 L_1 = \frac{C}{C_{OX,N}} = \frac{I_{M0}t}{V_{SPH}C_{OX,N}}. \quad \text{(Eq. 3.44)}$$

Now, Eq. 3.41 would be

$$A_{TOTAL} = (1um \times 350nm) + \frac{1um}{\left(\frac{V_{DD3A} - V_{SPH}}{V_{SPH} - V_{TN}}\right)^2}(350nm) + \frac{I_{M0}t}{V_{SPH}C_{OX,N}}. \quad \text{(Eq. 3.45)}$$



(a) (b)

Figure 3.36 (a) V_{SPH} vs area of M_1, (b) V_{SPH} vs area of M_{12}

Then, V_{SPH} has to be converted to V'_{SPH} by using Eq. 3.46

$$V_{SPH} = \frac{V'_{SPH} - \beta}{\alpha}. \quad \text{(Eq. 3.46)}$$

$$A_{TOTAL} = 350nm \times \left(1um + \frac{1um}{\left(\frac{V_{DD3A} - \frac{V'_{SPH} - \beta}{\alpha}}{\frac{V'_{SPH} - \beta}{\alpha} - V_{TN}} \right)^2} \right) + \frac{I_{M0} t}{\frac{V'_{SPH} - \beta}{\alpha} \times C_{OX,N}}. \quad \text{(Eq. 3.47)}$$

Now, there is only variable V_{SPH}. The size of M_1 and M_{12} are depicted in Fig. 3.36 and Fig. 3.37. The global optimum point was estimated as 2.79V. The required width of M_{12} was 50um for the optimum point, but the width of 20um was selected for balance of circuit, V_{SPH} of trade-off point is 2.76V.

Finally, the next job is to determine the capacitance of M_1. V_{SPH} of 2.76V is 83.6% of V_{DD3A}(3.3V). Then, the capacitance would be

$$t = \frac{VC}{I} = \frac{V_{SPH} C}{I_{M0}(t)}. \quad \text{(Eq. 3.48)}$$

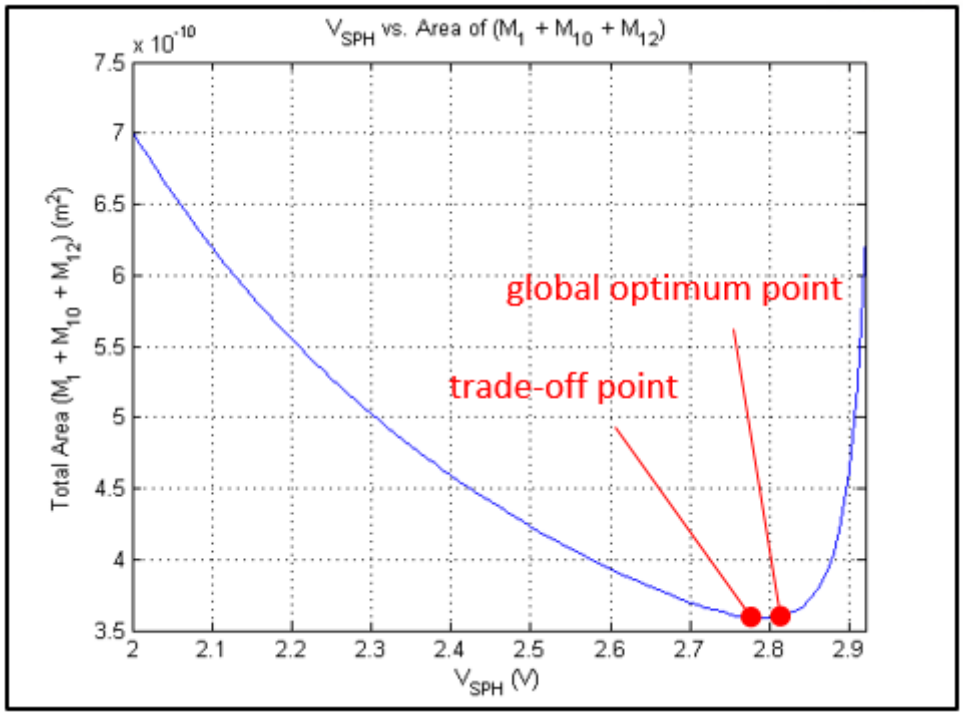
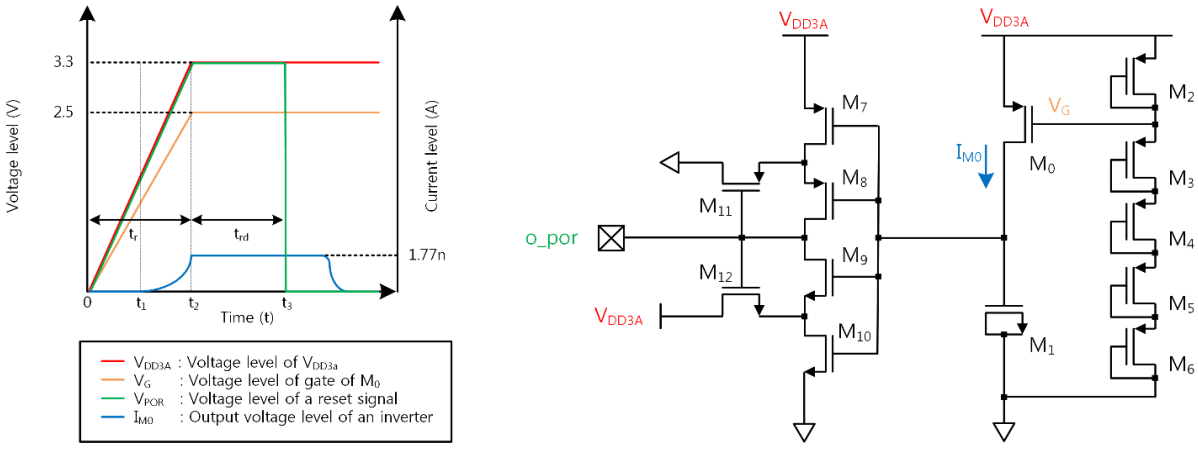


Figure 3.37 V_{SPH} vs total area (M_1 + M_{10} + M_{12})

$$C = \frac{It}{V} = \frac{\int_0^t I_{M0}(s)\ ds}{V_{SPH}}. \qquad \text{(Eq. 3.49)}$$

However, I_{M0} is a time-varying variable like Fig. 3.38 (a). The behavior of current I_{M0} can be modeled like Eq. 3.50.



(a) (b)

Figure 3.38 (a) Behavior of a designed PoR, (b) The full circuit of a designed PoR

$$\int_0^{t_3} I_{M0}(t)\,dt = \int_{t_1}^{t_2} \frac{1}{2}\mu_P C_{OX,P}\left(\frac{W_{M0}}{L_{M0}}\right)\left(|V_{GS}(t)| - |V_{TH,P}|\right)^2 dt + \int_{t_2}^{t_3} 1.77nA\,dt. \quad \text{(Eq. 3.50)}$$

V_{DD3A} can be modeled like Eq. 3.51.

$$V_{DD3A}(t) = \frac{3.3V}{1ms}\,t \times (u[t] - u[t-t_2]) + 3.3 \times u[t-t_2], \quad \text{(Eq. 3.51)}$$

where u[t] is unit step function.

Then, $V_G(t)$ and $V_S(t)$ is

$$V_G(t) = \frac{2.5}{3.3}V_{DD3A}(t). \quad \text{(Eq. 3.52)}$$

$$V_S(t) = V_{DD3A}(t). \quad \text{(Eq. 3.53)}$$

$$|V_{GS}(t)| = |V_G(t) - V_S(t)| = \left|\frac{2.5}{3.3}V_{DD3A}(t) - V_{DD3A}(t)\right| = \frac{0.8}{3.3}V_{DD3A}(t). \quad \text{(Eq. 3.54)}$$

Apply Eq. 3.54 to Eq. 3.50, then

$$\int_0^{t_3} I_{M0}(t)\,dt = \int_{t_1}^{t_2} 7.368 \times 10^{-7} \times \left(\frac{20um}{400nm}\right)\left(|V_{GS}(t)| - |V_{TH,P}|\right)^2 dt + (1.77nA \times t_{rd}) \quad \text{(Eq. 3.55)}$$

$$= \int_{t_1}^{t_2} K_P\left(\frac{0.8}{3.3}V_{DD3A}(t) - |V_{TH,P}|\right)^2 dt + (1.77nA \times 256us) \quad \text{(Eq. 3.56)}$$

$$= K_P \int_{t_1}^{t_2} \left(\frac{0.8}{3.3} \times \frac{3.3}{0.001}t - |V_{TH,P}|\right)^2 dt + 4.531 \times 10^{-13} \quad \text{(Eq. 3.57)}$$

$$= K_P \int_{t_1}^{t_2} \left(800t - \left|V_{TH,P}\right|\right)^2 \, dt + 4.531 \times 10^{-13} \quad \text{(Eq. 3.58)}$$

$$= K_P \int_{t_1}^{t_2} 640000t^2 - 1600\left|V_{TH,P}\right|t + {V_{TH,P}}^2 \, dt + 4.531 \times 10^{-13} \quad \text{(Eq. 3.59)}$$

$$= K_P \left[\frac{640000}{3}({t_2}^3 - {t_1}^3) - 800\left|V_{TH,P}\right|({t_2}^2 - {t_1}^2) + {V_{TH,P}}^2 \, (t_2 - t_1)\right] + 4.531 \times 10^{-13} \quad \text{(Eq. 3.60)}$$

t_1 is a point that M_0 is turned on, i.e., the point when $V_{GS} \geq |V_{TH,P}|$. Refer to Fig. 3.38 (a). $V_{TH,P}$ is 0.752V originally, but in this case it should handle the sub nA current. $V_{TH,P}$ is adjusted to 0.484V for the subthreshold compensation. t_1 could be calculated by Eq. 3.61.

$$|V_{GS}| \geq \left|V_{TH,P}\right| \quad \text{(Eq. 3.61)}$$

$$800t \geq 0.484 \quad \text{(Eq. 3.62)}$$

$$t \geq 605us \quad \text{(Eq. 3.63)}$$

$$\therefore t_1 \equiv 605us. \quad \text{(Eq. 3.64)}$$

By definition in Fig. 3.38 (a),

$$t_2 = t_r = 1ms. \quad \text{(Eq. 3.65)}$$

$$t_3 = t_r + t_{rd} = 1ms + 256us = 1.256ms. \quad \text{(Eq. 3.66)}$$

Then, put t_2 and t_2 to Eq. 3.60, the calculation result is

$$\int_0^{t_3} I_{M0}(t) \, dt = 6.467 \times 10^{-13} \, (C). \quad \text{(Eq. 3.67)}$$

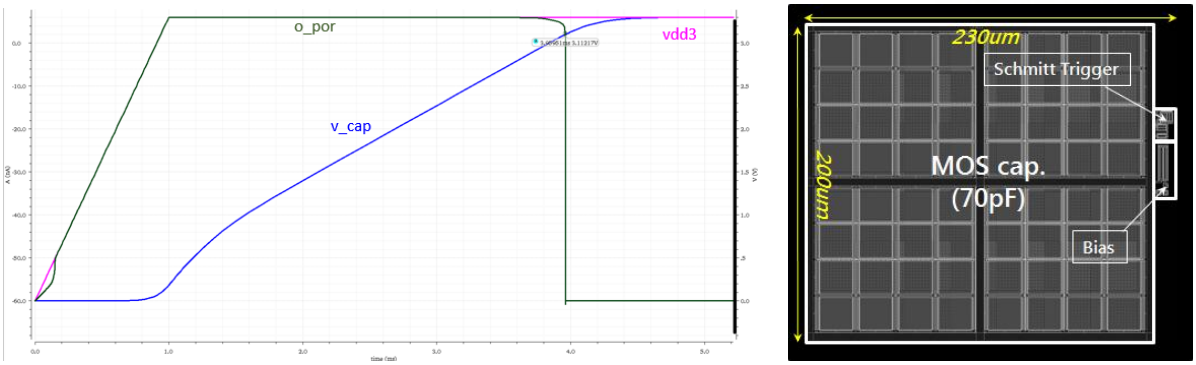
The required capacitance could be obtained by using Eq. 3.49.

$$C = \frac{6.467 \times 10^{-13}}{V_{SPH}} = 234.4 \, (fF). \quad \text{(Eq. 3.68)}$$

The size of the capacitor would be

$$W_1 L_1 = \frac{C}{C_{OX,N}} = \frac{234.4 \times 10^{-15}}{2.07 \times 10^{-3}} = 113.22 \, um^2. \quad \text{(Eq. 3.69)}$$

$$\sqrt{W_1 L_1} = 10.64 \, um. \quad \text{(Eq. 3.70)}$$



(a) (b)

Figure 3.39 (a) Simulation result of *PoR*, (b) Implementation result (layout)

The value of 10.64um is the minimum requirement for achieving 234.4fF, but the value of 11um was selected for an extra margin.

Fig. 3.39 shows the result of simulation. The typical reset duration is 2,500us (@tt, VDD3=3.3V, 27℃). The detailed reset duration depending on PVT variation is described in Table 3.11. Also, the tendency of PVT is depicted in Fig. 3.40.

Table 3.11 PVT simulation result (***PoR***)

Variation Item	Value	T_{rd} (μs)
Process	ss	23,713
	sf	1,873
	tt	4,276
	fs	24,763
	ff	1,861
Voltage	3.1	20,975
	3.3	8,511
	3.5	4,406
Temp	-40	33,215
	27	6,651
	100	3,046
	150	2,276

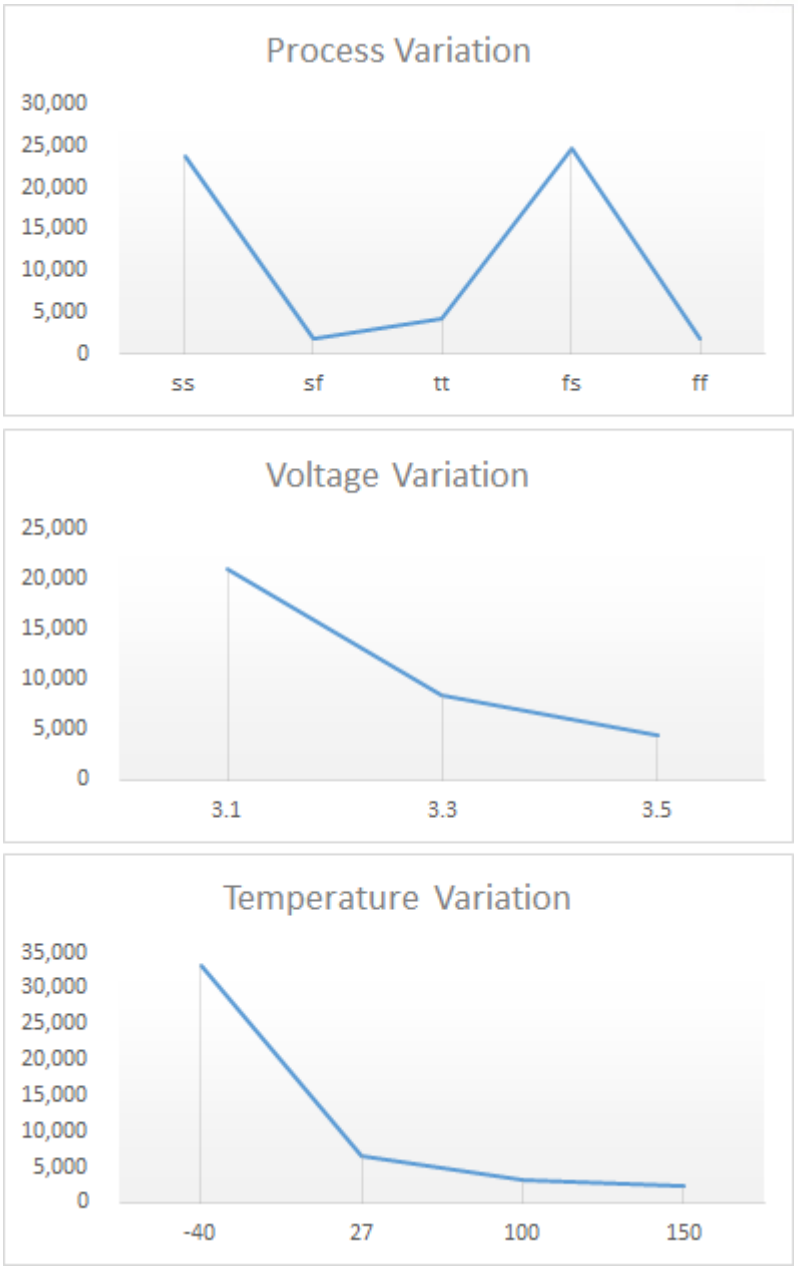


Figure 3.40 Reset duration variation tendency - PVT simulation (*PoR*)

3.2.2.9 *Digital Logic*

Digital Logic provides control signals for analog sub blocks. It controls block enable, current trimming for sub blocks. The operation mode (Normal and Low-power) is controlled by ***Digital Logic*** block. It supports protection techniques such as thermal shutdown, TXD dominant time-out and so on. The block also supports a remote wake-up function.

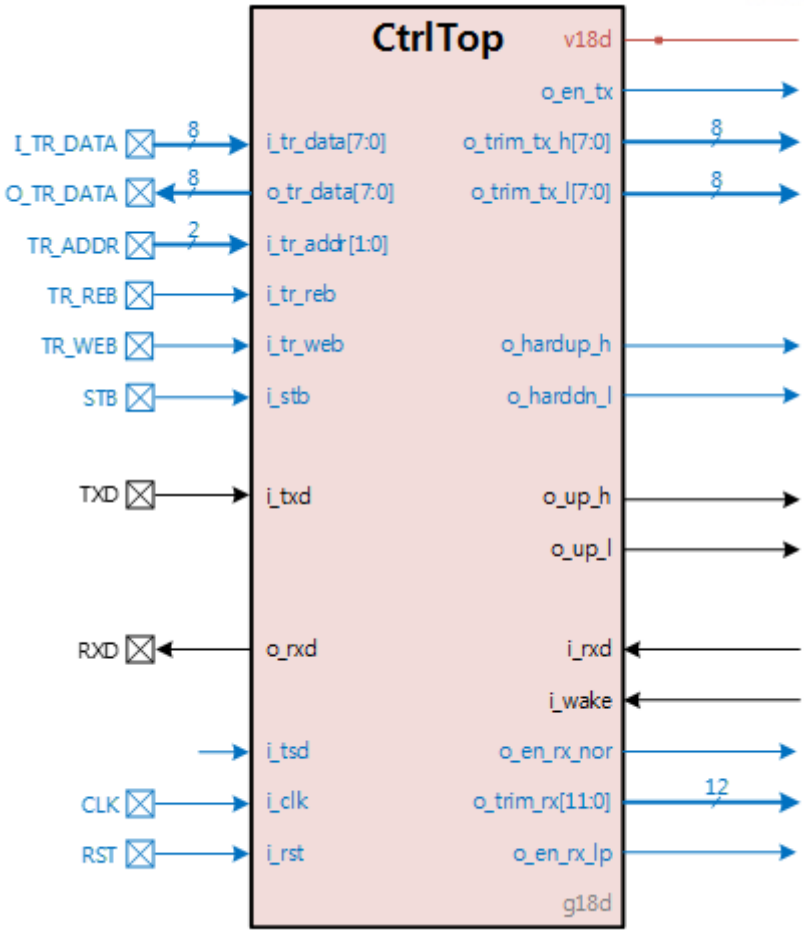


Figure 3.41 Block diagram of *Digital Logic*

Digital Logic receives the transmit-data bits via TXD pin, and converts it to driver-driving signals for a transmitter. Then, the transmitter (***Driver***) would be controlled by this control signals, and would load the analog data signals on the CAN bus lines.

Digital Logic has two different operating modes, Normal mode and Low-power mode. ***Digital Logic*** generates the block enable signals based on the operating mode. However, ***Digital Logic*** accepts the fault detection signals (thermal shutdown and TXD dominant time-out), and they also affect the block enable signals. ***Digital Logic*** supports the remote wake-up function, and performs the digital signal processing for overall transceiver functions.

Fig. 3.41 is a block diagram of ***Digital Logic***, and I/O configuration. Table 3.12 provides signal descriptions

Table 3.12 Signal descriptions (***Digital Logic***)

Signal name	Type	Description
v18d	POWER	1.8V digital supply voltage
g18d	POWER	Digital ground
i_tr_data[7:0]	IN	8-bit current trimming register file input
o_tr_data[7:0]	IN	8-bit current trimming register file output
i_tr_addr[1:0]	IN	2-bit current trimming register file address selection input
i_tr_reb	IN	Read enable signal for current trimming register file (active low)
i_tr_web	IN	Write enable signal for current trimming register file (active low)
i_stb	IN	Standby mode control input via pin STB
i_wake	IN	Remote wake-up signal generated from ***LPRCVR***
i_rxd	IN	Received data output from ***RCVR***; reads out data from the bus lines
i_tsd	IN	Over-temperature detection signal generated from ***TSD***
i_txd	IN	Transmit data input via pin TXD
i_clk	IN	4MHz clock input generated from ***OSC***
i_rst	IN	External asynchronous reset input (active high)
o_trim_tx_h[7:0]	OUT	8-bit current trimming output for ***Driver*** (CANH)
o_trim_tx_l[7:0]	OUT	8-bit current trimming output for ***Driver*** (CANL)
o_trim_rx[11:0]	OUT	12-bit current trimming output for ***RCVR*** and ***LPRCVR***
o_hardup_h	OUT	Hard pull-up for ***Driver*** (CANH)
o_harddn_l	OUT	Hard pull-down for ***Driver*** (CANL)
o_up_h	OUT	***Driver*** control signal to load the transmit bit (CANH)
o_up_l	OUT	***Driver*** control signal to load the transmit bit (CANL)
o_en_tx	OUT	Block enable signal for ***TxTop*** and its sub-blocks
o_en_rx_nor	OUT	Block enable signal for ***RCVR***
o_en_rx_lp	OUT	Block enable signal for ***LPRCVR***
o_rxd	OUT	Received data output from ***RCVR*** *(Normal mode)*; Remote wake-up output from ***LPRCVR*** (Standby mode) via pin RXD

Table 3.13 Operation mode

Mode	Pin STB	Pin RXD	
		Logical low	Logical high
Normal	Logical low	Bus dominant	Bus recessive
Standby	Logical high	Wake-up request detected	No wake-up request detected

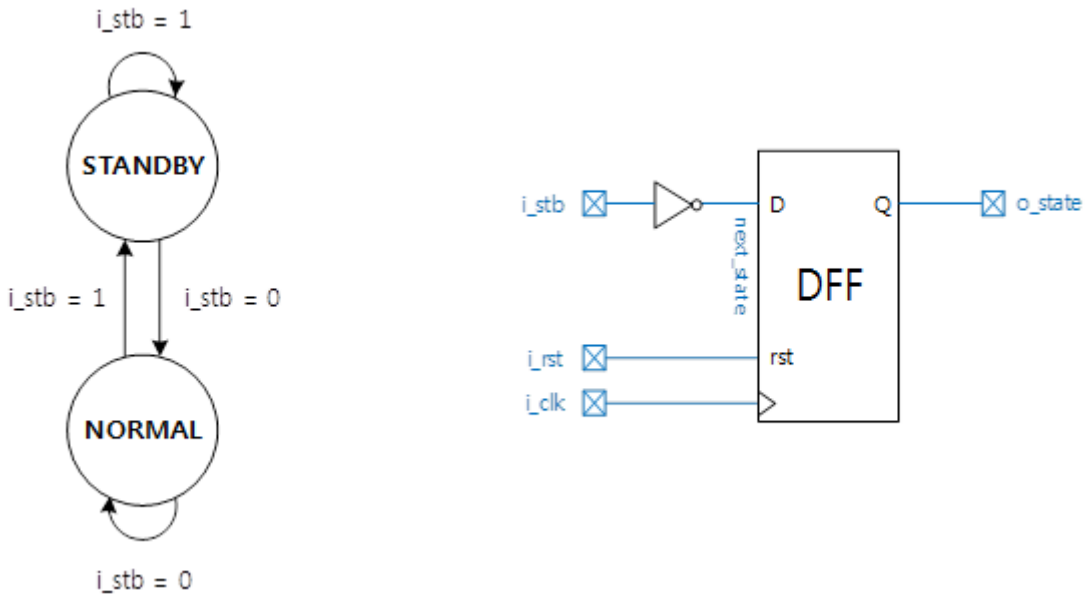
This module provides two modes of operation, which are selectable via pin STB. See Table 3.13 for a description of the modes of operation.

Digital Logic - FSM sub block

In Normal mode, the transceiver is able to transmit and receive data via the bus line (CANH/CANL). ***RCVR*** block converts the analog data from the bus lines into digital data which is output to pin RXD via the multiplexer (***RXDOutputSelector*** sub block). The slope of the output signal on the bus lines is tunable by current trimming.

In Standby mode, the transmitter and receiver are switched off, and ***LPRCVR*** will monitor the bus lines. A logical high on pin STB activates this ***LPRCVR*** and the wake-up filter (***WakeFlag*** sub block), and after t_{BUS} the state of the CAN bus is reflected on pin RXD for wake-up signal. Once the ***LPRCVR*** has detected a domain bus level for more than t_{BUS}, pin RXD will become logical low. The performance of ***LPRCVR*** is poor, so it can't recognize the bus level correctly sometimes. ***LPRCVR*** misidentifies the bus noise for dominant bit sometimes. Thus, wake-up filter is required, and it is ***WakeFlag*** block which is digital filter.

The FSM truth table is in Table 3.14. Table 3.15 is FSM state definition, the bit assignment for each state. Table 3.16 is the signal description explaining each pin.



(a) (b)

Figure 3.42 (a) FSM diagram, (b) Block diagram of *FSM*

Table 3.14 FSM truth table

Input	Output
i_stb	next_state
0	1
1	0

Table 3.15 FSM state definition

State	Bit
Standby	0 (default)
Normal	1

Table 3.16 Signal description (***FSM*** block)

Signal Name	Description
i_stb	Pin STB input
next_state	Input data of DFF (D register)
o_state	Current FSM state output
i_clk	System clock input
i_rst	Asynchronous reset signal input

Digital Logic - TSDFlag sub block

TSD flag signal (*o_tsd_flag*) will be set when *i_tsd* is logical high (1). However, the flag will be not cleared, even *i_tsd* goes to logical low (0). The flag only will be cleared when *i_tsd* is 0 and *i_txd* is 1. This is a kind of fail safe. This prevents output driver oscillation due to temperature drift.

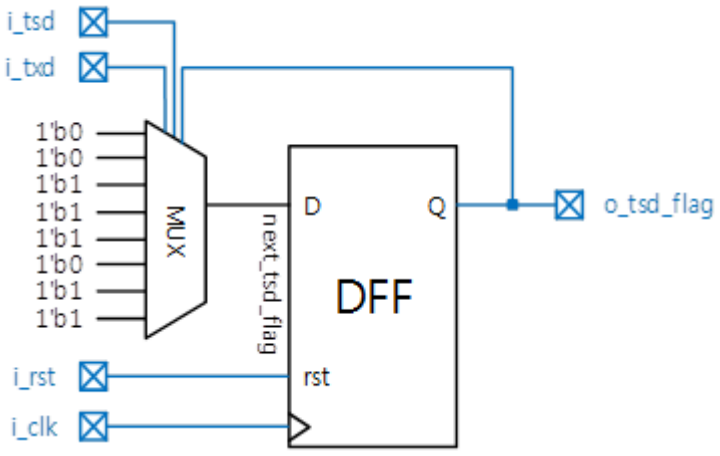


Figure 3.43 Block diagram of *TSDFlag*

Table 3.17 Truth table of ***TSDFlag*** block

Input			Output
o_tsd_flag	i_tsd	i_txd	next_tsd_flag
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Table 3.18 Signal description (***TSDFlag*** block)

Signal Name	Description
i_tsd	TSD block output; MUX selection signal
i_txd	Transmit data from MCU; MUX selection signal
next_tsd_flag	Input data of DFF (D register); MUX output
o_tsd_flag	TSD flag output; MUX selection signal
i_clk	System clock input
i_rst	Asynchronous reset signal input

Digital Logic - TXDTimeOutFlag sub block

If the duration of the dominant bit on pin TXD exceeds 600us, TXD dominant time-out block should be triggered. The system clock is 4MHz, and the period of a clock pulse is 0.25us.

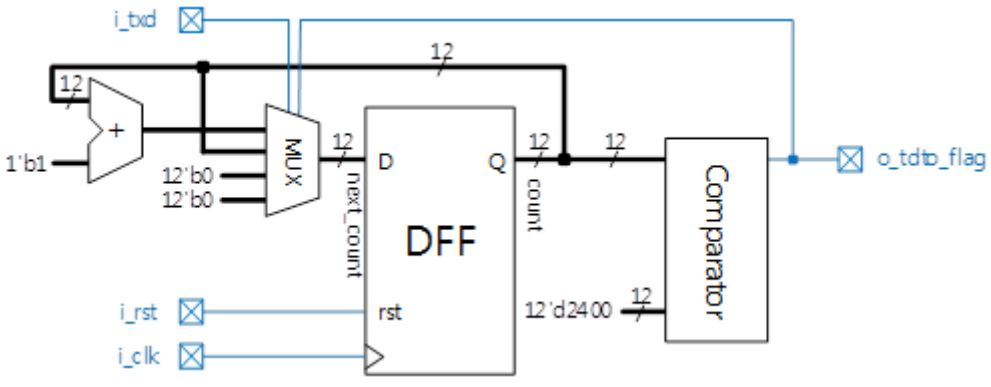


Figure 3.44 Block diagram of *TXDTimeOutFlag*

Table 3.19 Definition of *o_tdto_flag*

Input	Output
count	o_tdto_flag
count < 2400	0
count >= 2400	1

Table 3.20 Truth table for accumulator (***TXDTimeOutFlag*** block)

Input		Output
i_txd	o_tdto_flag	next_count
0	0	count + 1
0	1	count
1	x	0

Table 3.21 Signal description (***TXDTimeOutFlag*** block)

Signal Name	Description
i_txd	Transmit data from MCU; MUX select
count	Current counting; register output
next_count	Next counting; MUX output
o_tdto_flag	Active-high TXD dominant time-out flag

Therefore, the required number of clock pulse is 2400 for counting 600us. This can be implemented by using an accumulator and a comparator unit. This is defined as Table 3.19.

The accumulator will be triggered when *i_txd* is dominant bit (0). The accumulator stops counting the number of clock pulse, once counted number is larger than 2400. Then, *o_tdto_flag* becomes 1. The accumulator will be reset when *i_txd* is recessive bit (1), especially pos-edge of *i_txd*. However, it implemented as synchronous reset, not asynchronous reset. Because it is for robustness against noise at pin TXD.

Digital Logic - WakeFlag sub block

This is almost same with ***TXDTimeOutFlag*** bock. The only difference is that the counting number is 7. If the duration of dominant bit on *i_wake* exceeds 1.75us, the wake-up flag will be logical low. The system clock speed is 4MHz, and the period of a clock pulse is 0.25us. Therefore, the required number of clock pulse is 7 for 1.75us.

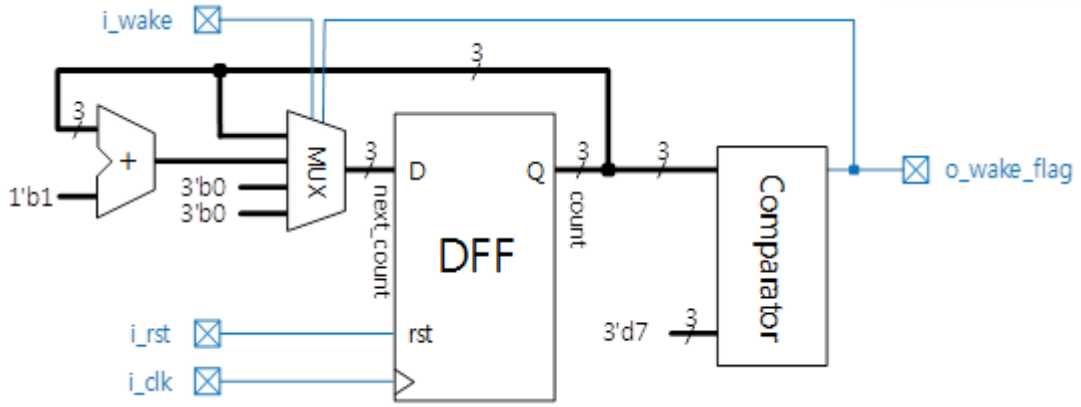


Figure 3.45 Block diagram of *WakeFlag*

The counter will be triggered when *i_wake* is dominant bit (0). The counter stops counting the number of clock pulse, once the counted number is larger than 7. Then *o_wake_flag* becomes 0. The counter will be reset immediately when *i_wake* goes back to recessive bit (1).

Table 3.22 Definition of *o_wake_flag*

Input	Output
Count	o_wake_flag
count < 7	1
count >= 7	0

Table 3.23 Truth table for counter (***WakeFlag*** block)

Input		Output
i_wake	o_wake_flag	next_count
0	0	count
0	1	count + 1
1	x	0

Table 3.24 Signal description (***WakeFlag*** block)

Signal Name	Description
i_wake	Wake-up signal input
count	Current counting; register output
next_count	Next counting; MUX output
o_wake_flag	Active-low wake-up timer flag

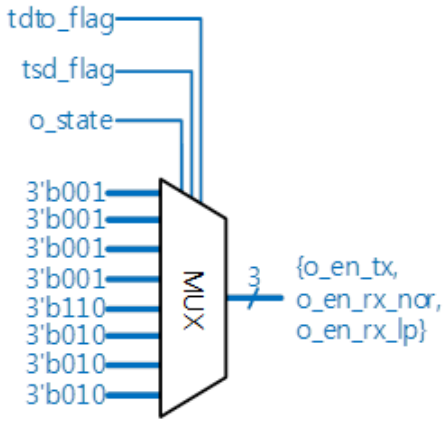


Figure 3.46 Block diagram of *EnableSignalGen*

Digital Logic – EnableSignalGen sub block

The transmitter should be disabled during *tsd_flag* or *tdto_flag* is logical high. The receiver should be alive for monitoring the state of bus.

Table 3.25 Truth table for block enable signals

Input			Output		
o_state	tsd_flag	tdto_flag	o_en_tx	o_en_rx_nor	o_en_rx_lp
Standby	0	0	0	0	1
Standby	0	1	0	0	1
Standby	1	0	0	0	1
Standby	1	1	0	0	1
Normal	0	0	1	1	0
Normal	0	1	0	1	0
Normal	1	0	0	1	0
Normal	1	1	0	1	0

Table 3.26 Signal description (***EnableSignalGen*** block)

Signal Name	Description
o_state	Current FSM state; FSM output
tsd_flag	Current TSD flag status; TSDFlag block output
tdto_flag	Current TXD dominant time-out flag status; TXDTimeOutFlag block output
o_en_tx	TX sub-blocks enable signals
o_en_rx_nor	RX normal receiver enable signal
o_en_rx_lp	RX low-power receiver enable signal

Digital Logic – TransmitterControl sub block

o_hardup_h and *o_harddn_l* are hard pull-up and pull-down control signal respectively. The hard pull-up (for CANH) and pull-down (for CANL) function would be activated by these two signal of *o_hardup_h* and *o_harddn_l*. It is activated when the transmitter is disabled. ***BI*** block will be forced to disengage from CAN bus lines during Standby mode or protection mode (TSD or TXD dominant time-out) by this hard pull-up and pull-down control. It leaks the charge from gate capacitor of switch MOS in ***BI*** block, so the switch MOS is forced to turned off. Table 3.27 shows the output values for the pull-up and pull-down control.

In Normal mode, the hard pull-up and pull-down function is not working.

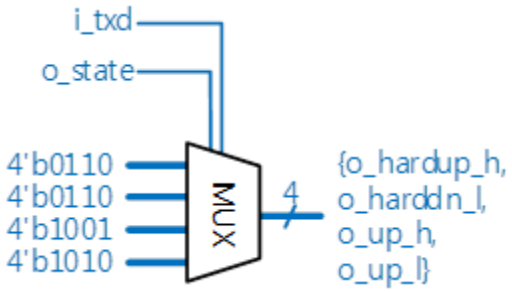


Figure 3.47 Block diagram of *TransmitterControl*

Table 3.27 Truth table for transmitter control signals

Input		Output			
i_tx_en	i_txd	o_hardup_h	o_harddn_l	o_up_h	o_up_l
0	x	0	1	1	0
1	0	1	0	0	1
1	1	1	0	1	0

Table 3.28 Signal description (***TransmitterControl*** block)

Signal Name	Description
i_tx_en	TX sub-blocks enable signal
i_txd	Transmit data from MCU; MUX selection signal
o_harddn_h	Hard pull-down output (for *Driver* block - CANH) (not available)
o_hardup_h	Hard pull-up output (for *Driver* block - CANH)
o_harddn_l	Hard pull-down output (for *Driver* block - CANL)
o_hardup_l	Hard pull-up output (for *Driver* block - CANL) (not available)
o_up_h	MOSFET driving signal for loading the data signal on bus lines (CANH)
o_up_l	MOSFET driving signal for loading the data signal on bus lines (CANL)

Digital Logic – RXDOutputSelector sub block

This function should be implemented by a multiplexer for preventing the unknown signal output. *i_rxd* signal is unknown state (i.e. electrically floating) during Standby mode because ***RCVR*** block is disabled. Thus its output is floating. Similarly, *i_wake* signal is also unknown state during Normal mode because ***LPRCVR*** is turned off.

Thus, the output signal line should be chosen by multiplexer. If this function is implemented by an OR-gate straightforwardly without this considerations, some unwanted situation will be occurred; for example, in Normal mode, *i_rxd* signal coming from deactivated ***RCVR*** becomes logical high (i.e. FLOATING→HIGH) temporarily due to some noise on wires, and the output of OR-gate will be logical high. As a result, *o_rxd* will be logical high, and it will be recognized as 'don't wake up', even though *i_wake* is logical low which means 'we have to wake up'

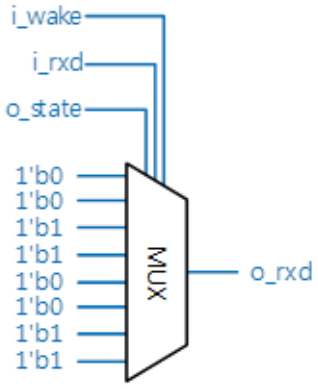


Figure 3.48 Block diagram of *RXDOutputSelector*

Table 3.29 Truth table for ***RXD output selector***

Input			Output
o_state	i_rxd	i_wake	o_rxd
Standby	x	0	0
Standby	x	1	1
Normal	0	x	0
Normal	1	x	1

Table 3.30 Signal description (***RXDOutputSelector*** block)

Signal Name	Description
o_state	Current FSM state; FSM output
i_rxd	RX data input coming from *RCVR* of *RxTop*
i_wake	Wake-up signal input coming from *LPRCVR* of *RxTop*
o_rxd	Selected output signal of MUX

Digital Logic – CurrentTrimmingRegFile sub block

This register file stores current trimming values. It controls ***Driver***, ***RCVR***, ***LPRCVR***, and ***OSC*** blocks. *i_tr_data* and *o_tr_data* is for current trimming. The initial values were pre-defined in register file, so it would be initialized by these default value. The default value is described in Table 3.32.

TX_H is for the high side of ***Driver***, TX_L for the low side of ***Driver***. Both of them has 8-bit width. RX_LP is for ***LPRCVR***, and it has 5-bit width. RX_CP is for a comparator of ***RCVR***, it has 3-bit width. RX_OFS is for ***RCVR***, it has 4-bit width. Finally, OSC is for an internal 4MHz oscillator (***OSC*** block), it has 4-bit width.

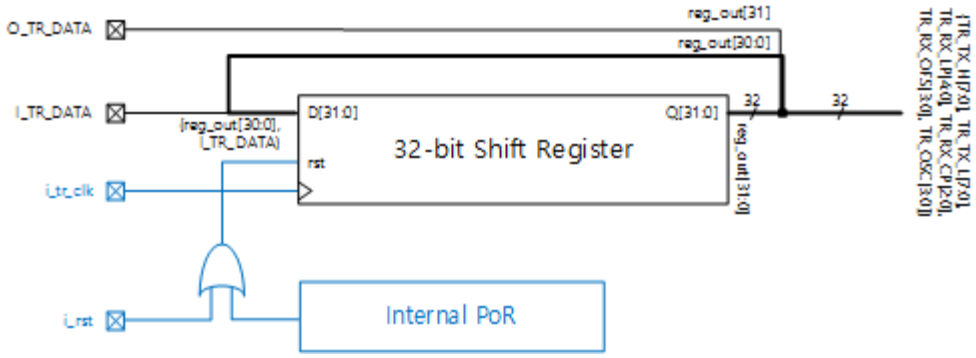


Figure 3.49 Block diagram of *CurrentTrimmingRegFile*

Table 3.31 Signal description (***CurrentTrimmingRegFile*** block)

Signal name	Description
i_tr_data	Shift register input; a single bit input of binary sequence of current trimming
i_tr_clk	Clock input for shift register

Table 3.32 Register assignment for current trimming data

Register	REG[31:24]	REG[23:16]	REG[15:8]	REG[7:0]
Trim data	TX_H	TX_L	{RX_LP, RX_CP}	{RX_OFS, OSC}
Init. value	0x78	0x7A	0x85	0x39

The RTL was synthesized, and P&R result is the following. The gate count was estimated about 1,000 gates. The timing margin was 198ns (@5MHz operating frequency). The implementation result is depicted in Fig. 3.50. The width is 145um, and height is 47um.

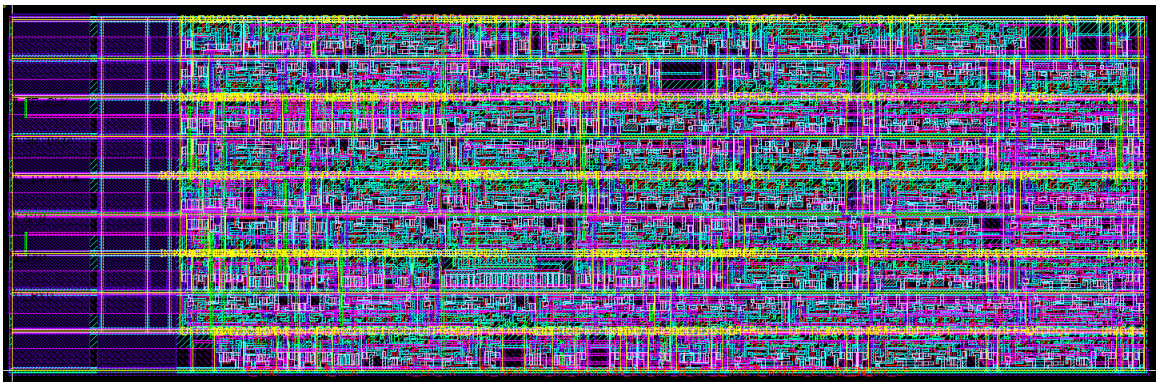


Figure 3.50 P&R result of *Digital Logic* block

3.2.3 Measurement Results

The transceiver was fabricated by 0.18um BCDMOS technology. Fig. 3.51 is the fabricated chip. The chip was packaged as 44-pin LQFP (Fig. 3.52). Fig. 3.53 is assembled for experimental measurement.

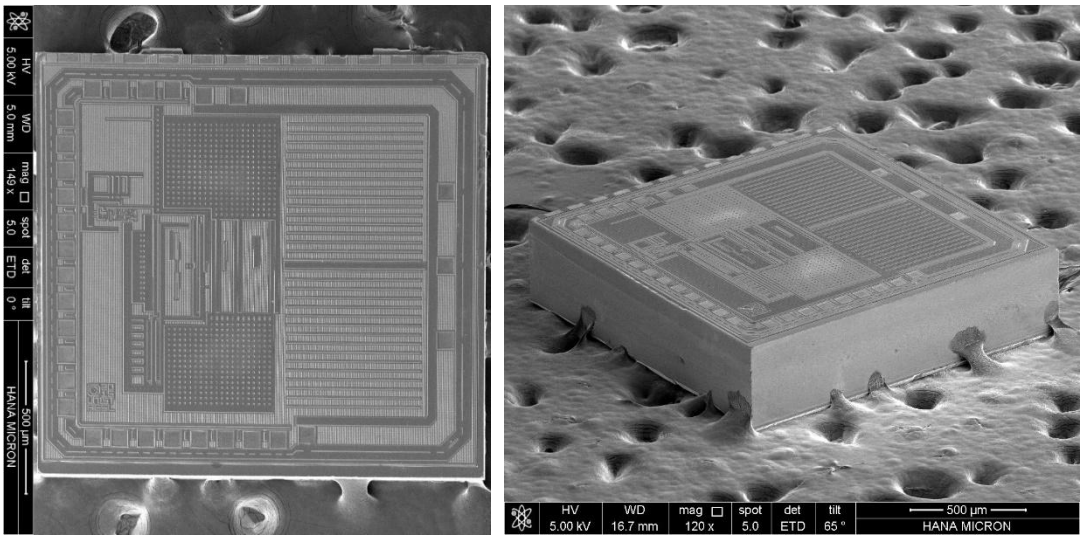


Figure 3.51 Die image of the CAN transceiver (magnification: x120)

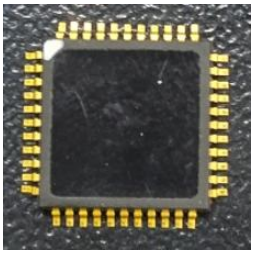


Figure 3.52 Packaged chip (44-pin LQFP)

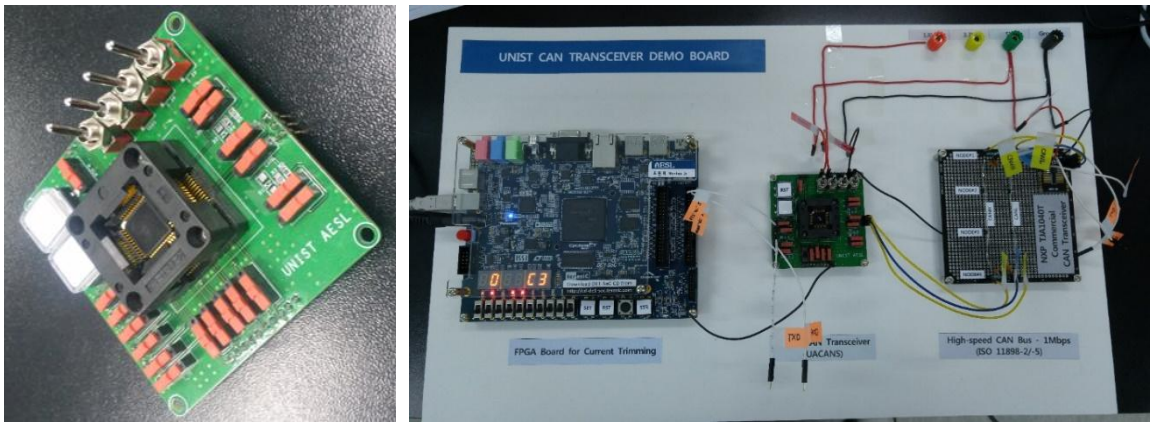


Figure 3.53 Test PCB and performance test environment

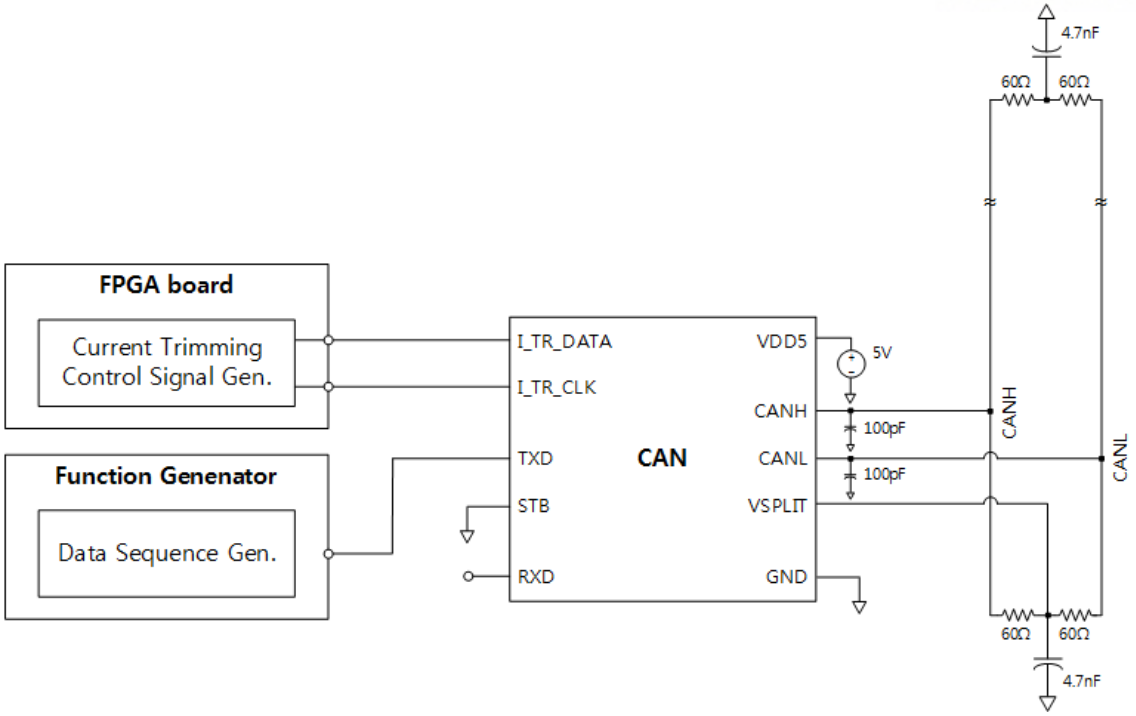


Figure 3.54 Typical environment for measurement

Fig. 3.54 shows a typical bus configuration for high-speed CAN. It also presents how to tune the transceiver by using current trimming. FPGA board is the tuner, it injects current trimming signal to transceiver via pin *I_TR_DATA* and *I_TR_CLK*.

The register file is implemented by shift register. *I_TR_DATA* is single input data for shift register. *I_TR_CLK* is clock input for shift register. The FPGA was developed for generating current trimming sequence and clock. The timing diagram is depicted in Fig. 3.55.

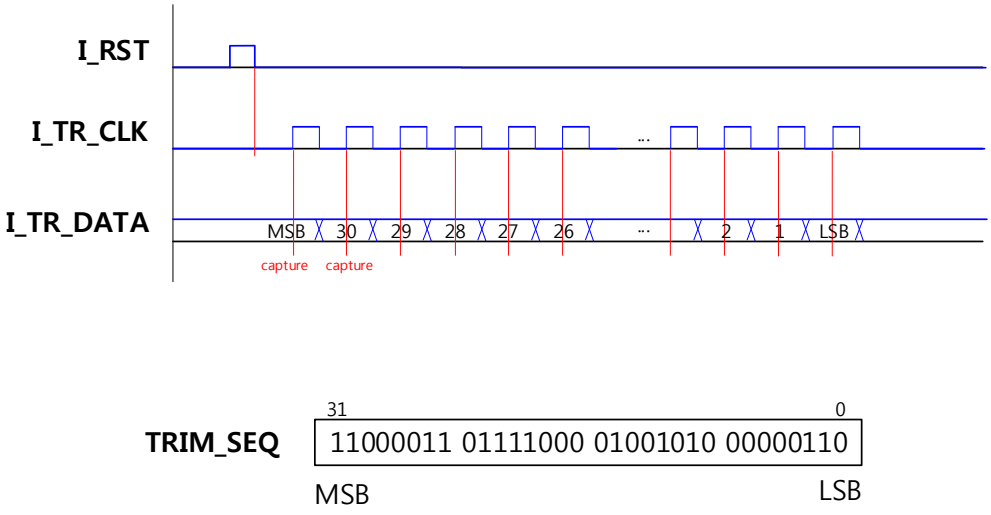
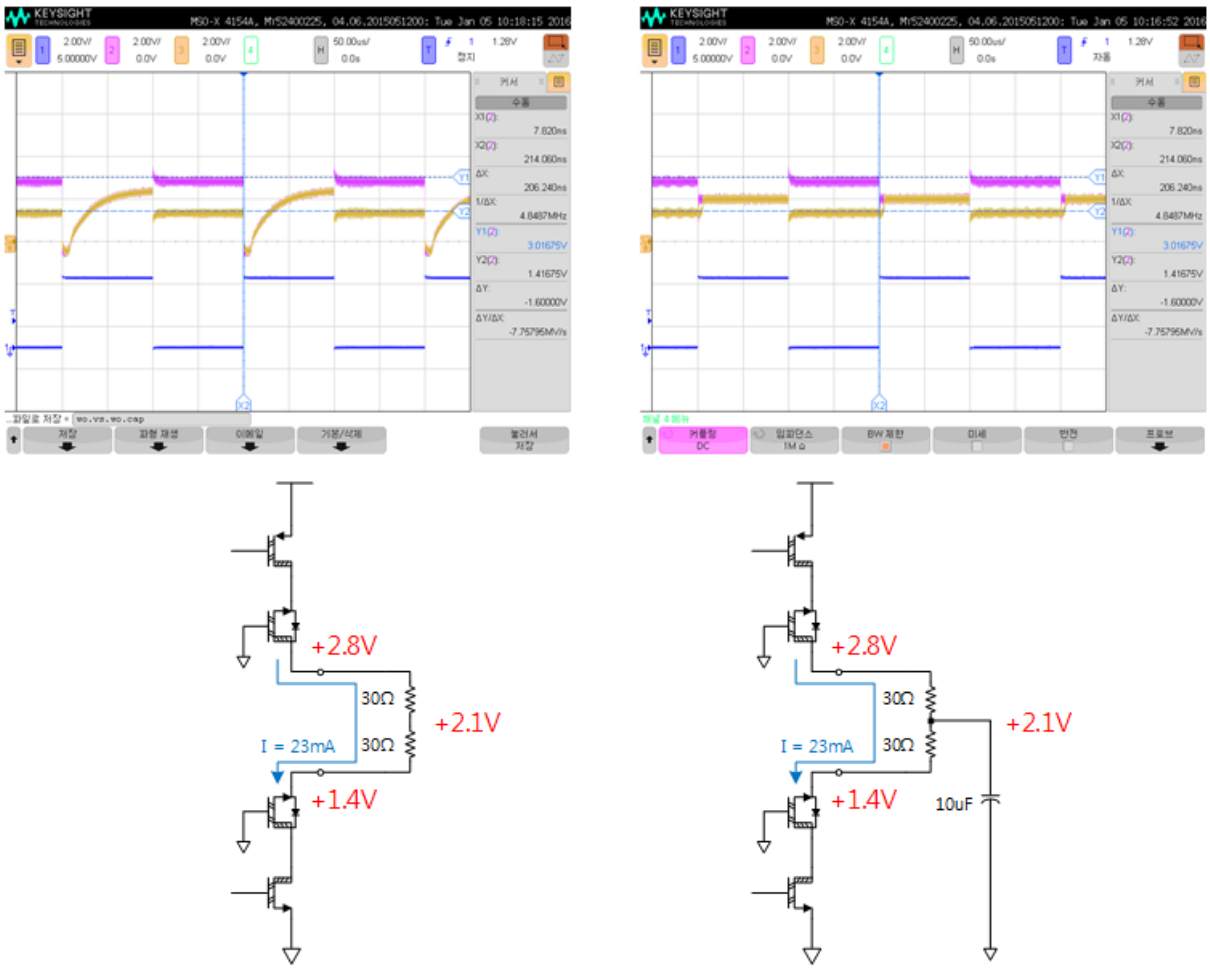


Figure 3.55 Timing diagram for current trimming

3.2.3.1 Output stage (*BI* and *VSplit*)

The Fig. 3.56 shows the measurement result of TX path (***BI*** and ***VSplit***). The bus is unstable when there is only bus termination resistor, Fig. 3.56 (a). The bus is discharged to ground level and slowly charged to 2.5V. This problem came from timing mismatch of CANH/CANL signal.

The switch MOS of CANL is turned off slightly later after CANH is turned off when the bus is transmitting recessive bit. The MOS of CANL is still turned on when the MOS of CANH is turned off, so the bus level is pulled down to ground. After then, both of CANH and CANL are turned off, the bus goes to floating state. Because both of switch MOS were turned off, so the impedance is almost infinitely high. However, the impedance values of turned off switches are same, so the bus level goes up to center point of 2.5V slowly.



(a) (b)

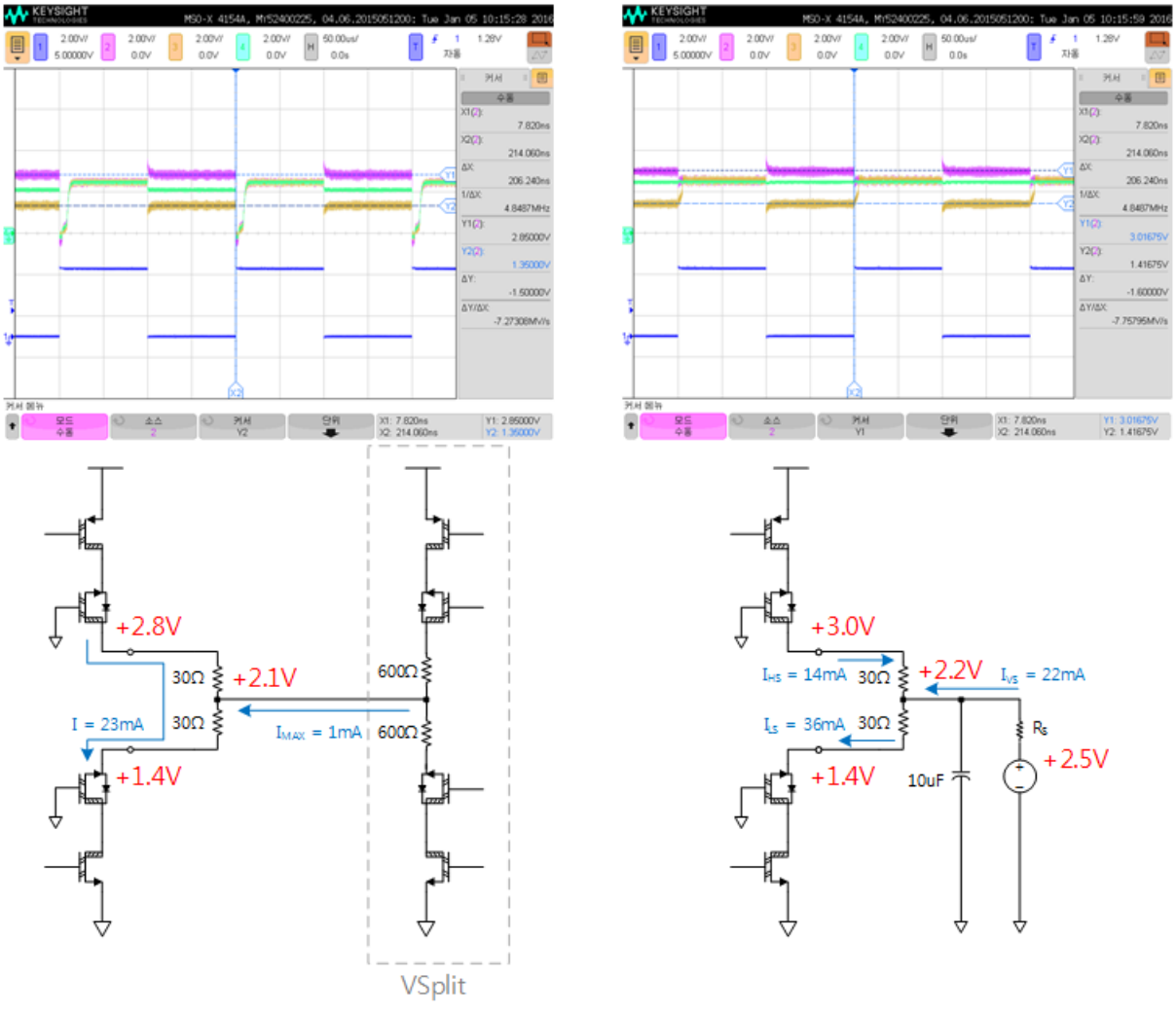
Figure 3.56 Output stage measurement result

(a) Measurement with only termination resistors, (b) 10uF capacitor is added at VSplit node

This phenomenon could be eliminated by adding capacitor at center point (VSplit node). Refer to Fig. 3.56 (b). The capacitor maintains the voltage level, even it is discharged very moment. It can be observed that CANL signal goes to the center point later than CANH signal.

In Fig. 3.56 (c), ***VSplit*** output is attached at center point node. ***VSplit*** supplies the current as a voltage source, but the current is limited to 1.5mA. The supply energy is so small, as a result, the bus is still unstable. Therefore, it is less effective. Refer to Table 3.33 for the details of ***VSplit***.

In Fig. 3.56 (d), the output of power supply was connected to center point. It supplies current almost infinitely, so center point could be preserved.

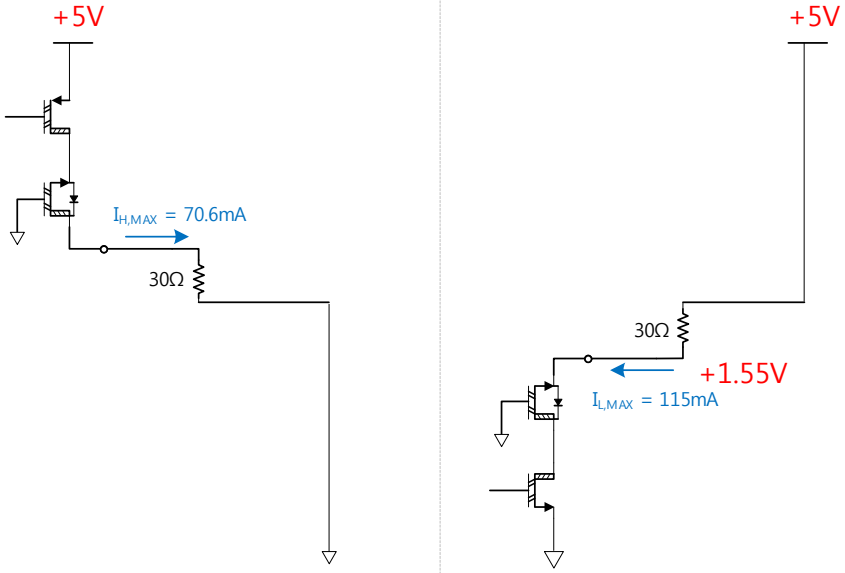


(c) (d)

Figure 3.56 (continued) (c) Output of VSplit is added (without 10uF capacitor), (d) Ideal voltage source is added instead of VSplit (with 10uF capacitor)

The timing mismatch of bus output might be caused by latch up phenomenon. Be careful when LDMOS was employed in the design in order to exploit the parasitic diode.

Fig. 3.57 shows current mismatching. In this case, PMOS switch has to be grown, or NMOS has to be shrunk for same current driving capability.



(a) (b)

Figure 3.57 Current matching experiment (a) high side test, (b) low side test

Table 3.33 Measurements of ***VSplit*** block

Test condition			Simulation value (tt)		Experimental value	
R_{LOAD}	Temp (℃)	V5A (V)	O_V_SPLIT(V)	I_{SPLIT}(μA)	O_V_SPLIT(V)	I_{SPLIT}(μA)
#1. Loaded & R_{LOAD}=5KΩ Connected with G5A	23.3	4.5	2.080	416.700	1.900	379.000
		5	2.190	438.900	2.040	400.000
		5.25	2.310	461.100	2.100	419.300
#2. Loaded & R_{LOAD}=5KΩ Connected with V5A		4.5	2.630	423.600	2.800	392.600
		5	2.770	446.000	2.900	413.400
		5.25	2.910	468.400	3.040	434.400
#3. Unloaded		4.5	2.33	N/A	2.3	N/A
		5	2.48		2.5	
		5.25	2.58		2.6	

3.2.3.2 Receivers - Normal receiver (*RCVR*) and Low-power receiver (*LPRCVR*)

The Fig. 3.59 is the measurement results of receivers. NXP's commercial CAN transceiver is used for performance comparison. TX signal is transmitted by NXP's one. The receiver output of the transceiver which is implemented in this work is depicted as AESL_RXD and AESL_WAKE in Fig. 3.59. The data rate was 1Mbps.

Fig. 3.59 (a) is the result of ***RCVR*** block. The timing of AESL_RXD signal is almost same with NXP_RXD. Fig. 3.59 (b) is the result of ***LPRCVR*** block. The wake-up signal of AESL_WAKE is risen up after 1.56us. This means that the both of ***LPRCVR*** and the wake-up filter of ***Digital Logic*** is working properly. However, the target was 1.75us, but the result is 1.56us due to irregular clock frequency of ***OSC*** block. The clock frequency was 4.48MHz.

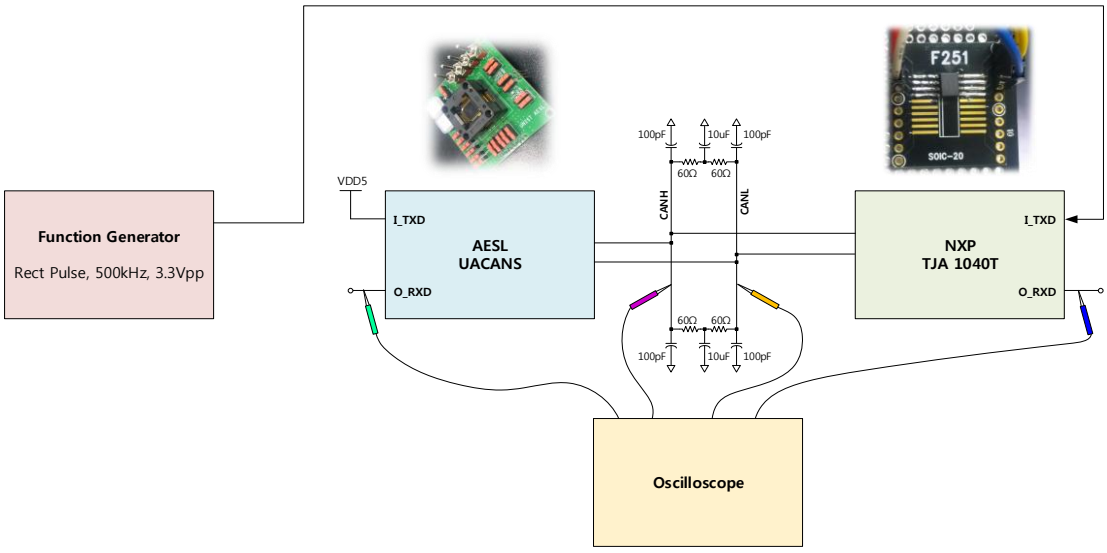
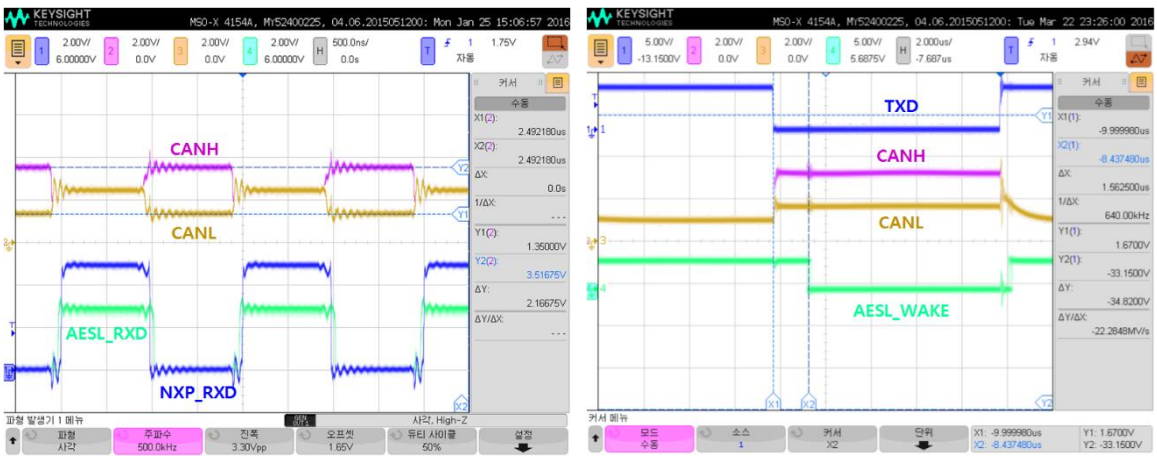


Figure 3.58 Test environment for receivers



(a) (b)

Figure 3.59 Measurement results (a) *RCVR* block, (B) *LPRCVR* block

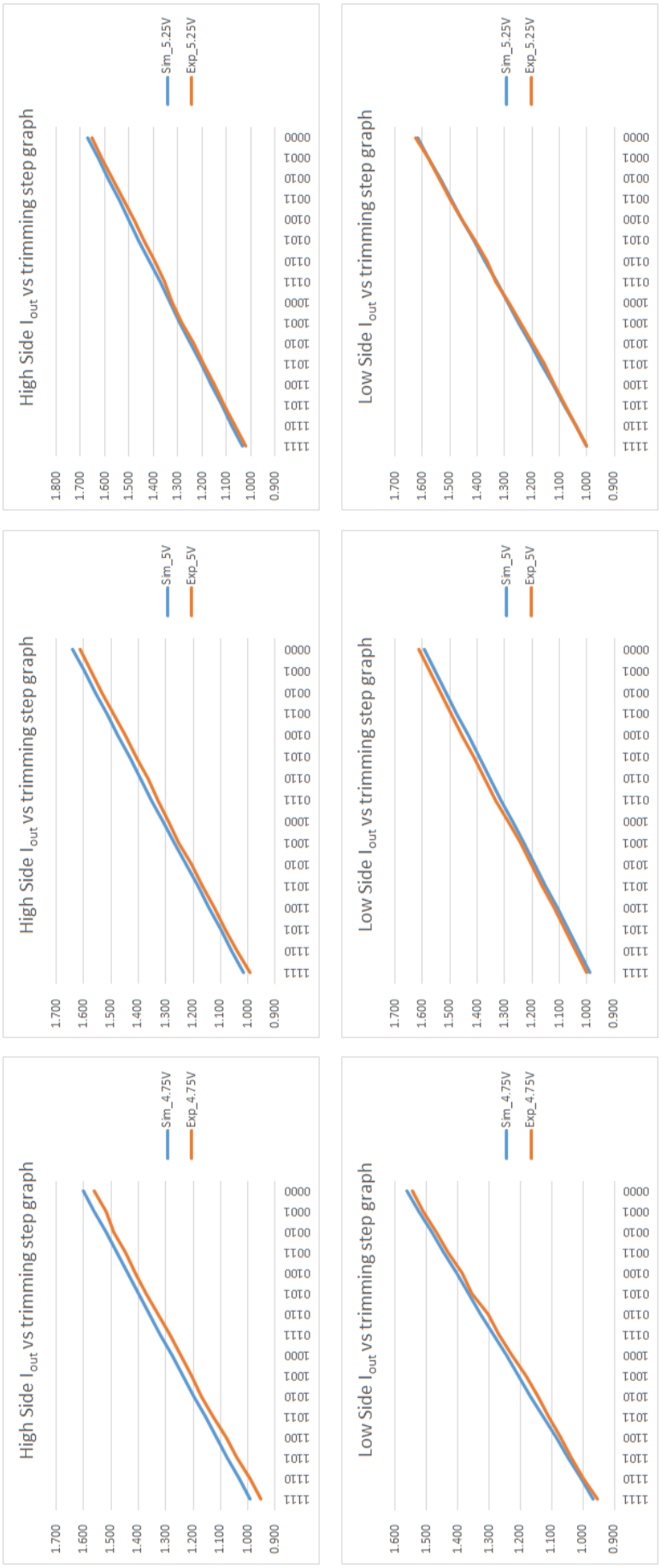
3.2.3.3 *Driver* block

The test condition is the following. The detailed measurement is presented in Table 3.34. Fig. 3.60 shows comparison between simulation value and measurement value.

- Temperature: 27℃
- Bias current (i_b_10u): 10uA
- R_{Load}: 1kΩ

Table 3.34 Current output measurement of ***Driver*** block with current trimming

Test condition		Simulation value (mA)			Experimental value (mA)		
Test target	i_trim	4.75V	5V	5.25V	4.75V	5V	5.25V
High side of ***Driver***	1111	0.993	1.017	1.035	0.952	0.991	1.019
	1110	1.033	1.059	1.078	0.992	1.036	1.060
	1101	1.075	1.101	1.121	1.040	1.078	1.108
	1100	1.115	1.142	1.163	1.080	1.120	1.151
	1011	1.156	1.184	1.206	1.128	1.165	1.193
	1010	1.197	1.226	1.248	1.169	1.210	1.237
	1001	1.237	1.268	1.291	1.208	1.255	1.283
	1000	1.278	1.309	1.333	1.244	1.293	1.328
	0111	1.319	1.351	1.375	1.286	1.329	1.360
	0110	1.359	1.392	1.417	1.329	1.373	1.409
	0101	1.399	1.433	1.460	1.373	1.420	1.452
	0100	1.439	1.474	1.502	1.410	1.456	1.492
	0011	1.480	1.516	1.544	1.448	1.492	1.530
	0010	1.520	1.557	1.585	1.490	1.535	1.570
	0001	1.560	1.598	1.627	1.520	1.570	1.611
	0000	1.600	1.639	1.669	1.560	1.610	1.650
Low side of ***Driver***	0000	0.967	0.988	1.001	0.954	1.000	1.000
	0001	1.007	1.028	1.042	1.000	1.039	1.039
	0010	1.047	1.069	1.084	1.037	1.082	1.082
	0011	1.087	1.109	1.125	1.072	1.121	1.121
	0100	1.127	1.150	1.166	1.109	1.164	1.156
	0101	1.167	1.190	1.207	1.146	1.204	1.200
	0110	1.207	1.231	1.249	1.183	1.244	1.244
	0111	1.246	1.272	1.290	1.227	1.284	1.284
	1000	1.286	1.313	1.331	1.269	1.332	1.334
	1001	1.326	1.353	1.372	1.305	1.371	1.366
	1010	1.366	1.394	1.413	1.354	1.414	1.410
	1011	1.405	1.434	1.454	1.385	1.454	1.454
	1100	1.445	1.475	1.495	1.430	1.495	1.498
	1101	1.484	1.515	1.536	1.471	1.536	1.538
	1110	1.524	1.556	1.578	1.512	1.575	1.580
	1111	1.564	1.596	1.618	1.547	1.615	1.624



(a) sample 1
(b) sample 2
(c) sample 3

Figure 3.60 Current output measurement (graphic) of *Driver* block

3.2.3.4 Thermal shutdown (*TSD*)

TSD block didn't work properly, the measurement is not available. ***TSD*** block should be revised in future wok.

3.2.3.5 Power-on Rest (*PoR*)

Fig. 3.61 shows the measurement of ***PoR*** block. The reset duration t_{rd} is measured by 2,500us. The target design value was 2,500us. It is verified successfully.

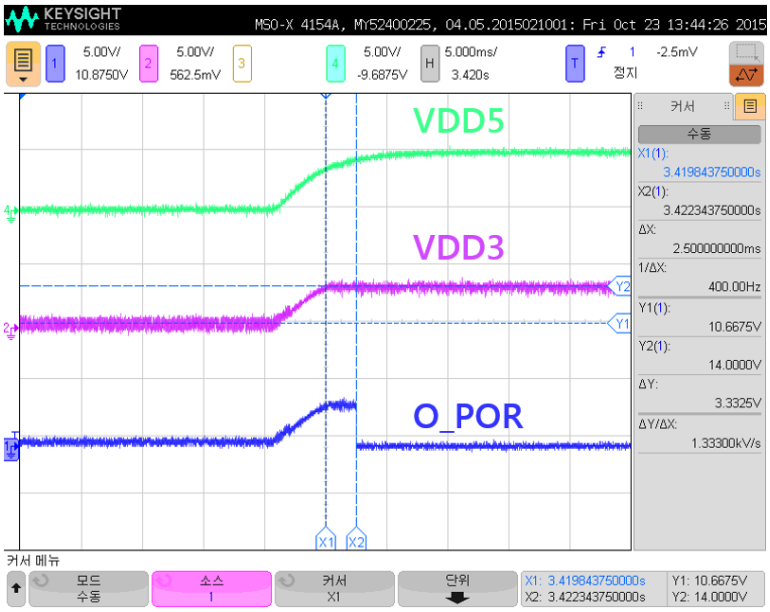


Figure 3.61 Performance measurement (*PoR*)

3.2.3.6 *Digital Logic* block

Table 3.35 shows the list of test vector, and the experimental result. All of the test vector were passed.

Table 3.35 Test vectors for ***Digital Logic*** block

Test order	Test vector	Result
1	FSM (Low-power mode)	✓
2	Thermal Shutdown	✓
3	Wake-up output (filter)	✓
4	TX-path (Driver control)	✓
5	RX-path (Receiver selector)	✓
6	Current Trimming Registers	✓

3.2.3.7 High Voltage Test

In this work, only low voltage ESD I/O pad was available. There is no high voltage I/O/ Thus, the used I/O pad was not the high voltage I/O pad. The structure of the pad is presented in Fig. 3.62. *O_CANH* pin is connected to V5A by an ESD diode. The high voltage of +40V is coming from *O_CANH* pin, and it goes to V5A. Therefore, V5A is affected by *O_CANH*, and this operation should not happen.

As a result, if the level of *O_CANH* is +40V, then V5A goes to +39.3V, and the PMOS will be damaged. Because the V_{GS} breakdown voltage of PMOS was 6V. Thus, high voltage measurement was canceled. Now, the high voltage I/O is available, and a revised CAN transceiver will be fabricated soon. At that time, high voltage performance will be measured.

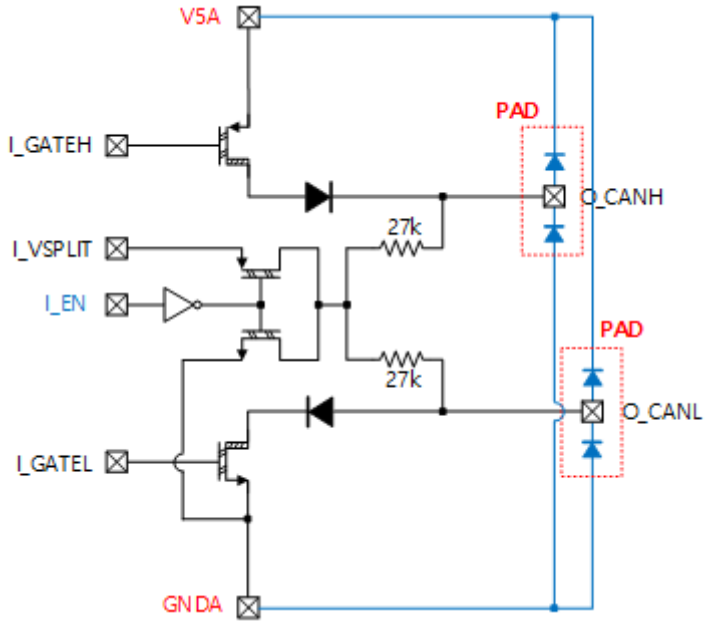


Figure 3.62 Pad issue on CANH and CANL pins

3.2.3.8 Operating Temperature Variation Test

The transceiver worked properly in wide temperature range from -40℃ to 150℃.

3.3 CAN Transceiver Type II: MCU Integrated CAN Transceiver

3.3.1 Architecture

The designed CAN transceiver was also embedded with a MCU as shown in Fig. 3.63. Intel 8051 compatible low-power MCU is used for this implementation. It has an integrated CAN controller which controls the CAN transceiver.

The main difference between the stand-alone version CAN transceiver (***Type I***) and the MCU version CAN transceiver (***Type II***) is a digital logic, especially ***CurrentTrimmingRegFile*** block. The CAN transceiver had to be connected to MCU bus for chip integration. Therefore, the block should support the bus interface for programming the register file of ***CurrentTrimmingRegFile*** block. The detailed description will go on in Section 3.3.2.

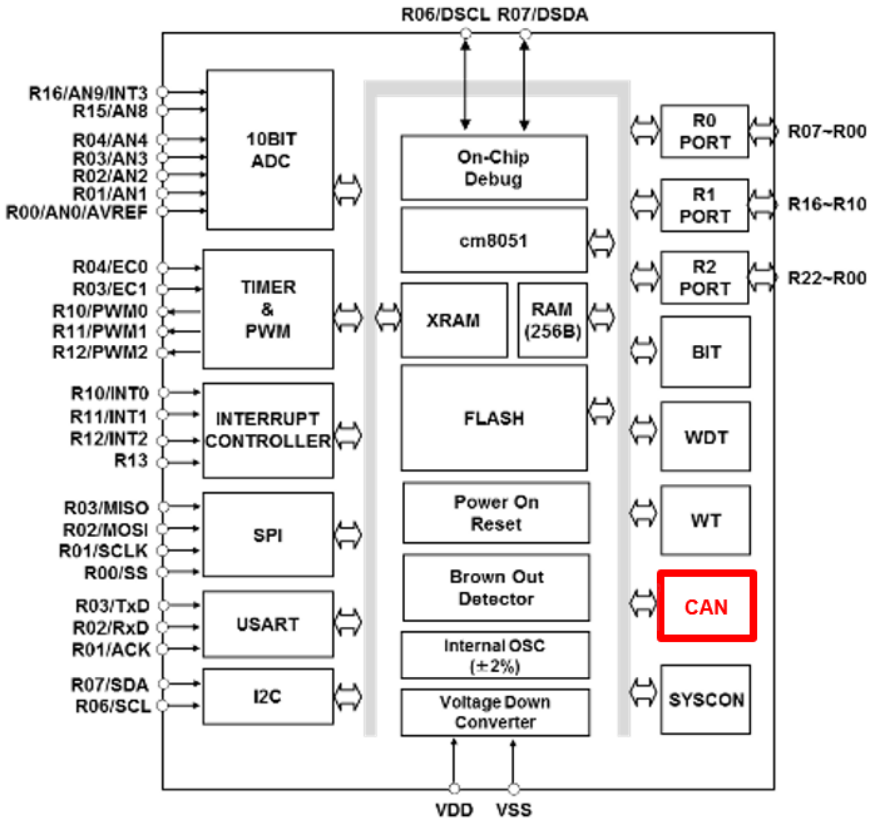


Figure 3.63Architecture of MCU integrated CAN transceiver (*Type II*)

3.3.2 Design and Implementation

The final implementation is shown in Fig. 3.64. The, and the transceiver core is in Fig. 3.65.

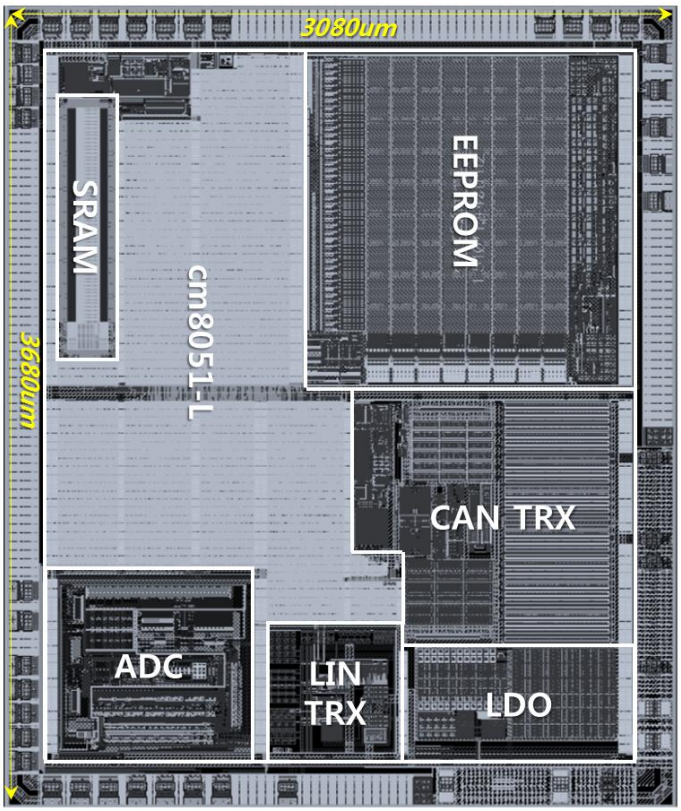


Figure 3.64 Layout of MCU integrated CAN transceiver (*Type II*)

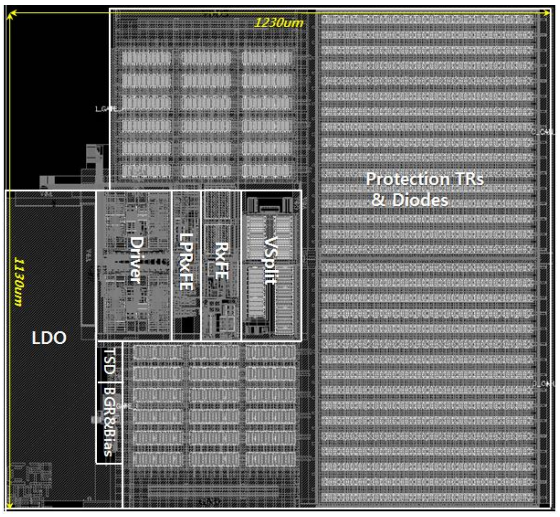


Figure 3.65 Layout of CAN transceiver core for *Type II*

3.3.2.1 *Digital Logic* block

Digital Logic – CurrentTrimmingRegFile sub block

This register file stores current trimming values. It controls ***Driver***, ***RCVR***, ***LPRCVR***, and ***OSC*** blocks. *i_tr_data* and *o_tr_data* have 8-bit width respectively, and they are connected to MCU's bus. The initial values were pre-defined in register file, so it would be initialized by these default value. The default value is described in Table 3.38.

TX_H is for the high side of ***Driver***, *TX_L* for the low side of ***Driver***. Both of them has 8-bit width. *RX_LP* is for ***LPRCVR***, and it has 5-bit width. *RX_CP* is for a comparator of ***RCVR***, it has 3-bit width. *RX_OFS* is for ***RCVR***, it has 4-bit width. Finally, *OSC* is for an internal 4MHz oscillator (***OSC*** block), it has 4-bit width.

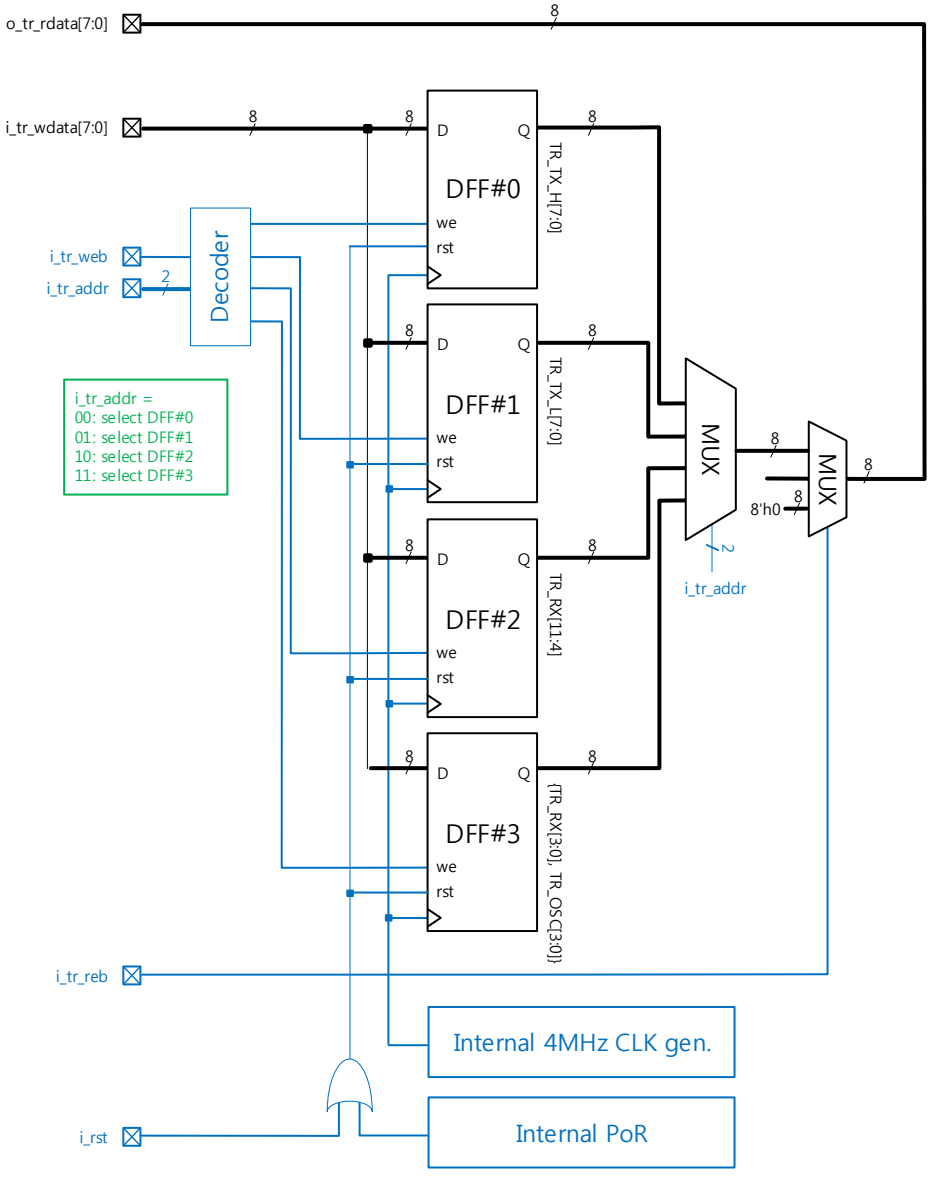


Figure 3.66 Block diagram of *CurrentTrimmingRegFile*

Table 3.36 Truth table for register addressing

Input		Register (8-bit width)			
i_tr_addr[1]	i_tr_addr[0]	REG#0	REG#1	REG#2	REG#3
0	0	O	X	X	X
0	1	X	O	X	X
1	0	X	X	O	X
1	1	X	X	X	O

Table 3.37 Signal description (***CurrentTrimmingRegFile*** block for ***Type II***)

Condition	Description
i_tr_reb = 0	Read mode. The value of a selected register would be shown on 'i_tr_data'.
i_tr_web = 0	Write mode. The value of a selected register would be updated by 'i_tr_data'.

Table 3.38 Register assignment for current trimming data

Register	REG#0	REG#1	REG#2	REG#3
Trim data	TX_H	TX_L	{RX_LP, RX_CP}	{RX_OFS, OSC}
Init. value	0x78	0x7A	0x85	0x39

3.3.3 Measurement Results

Fig. 3.67 is test environment, and Fig. 3.68 is the measurement result for each data rate; 1Mbps, 500kbps, 50kbps.

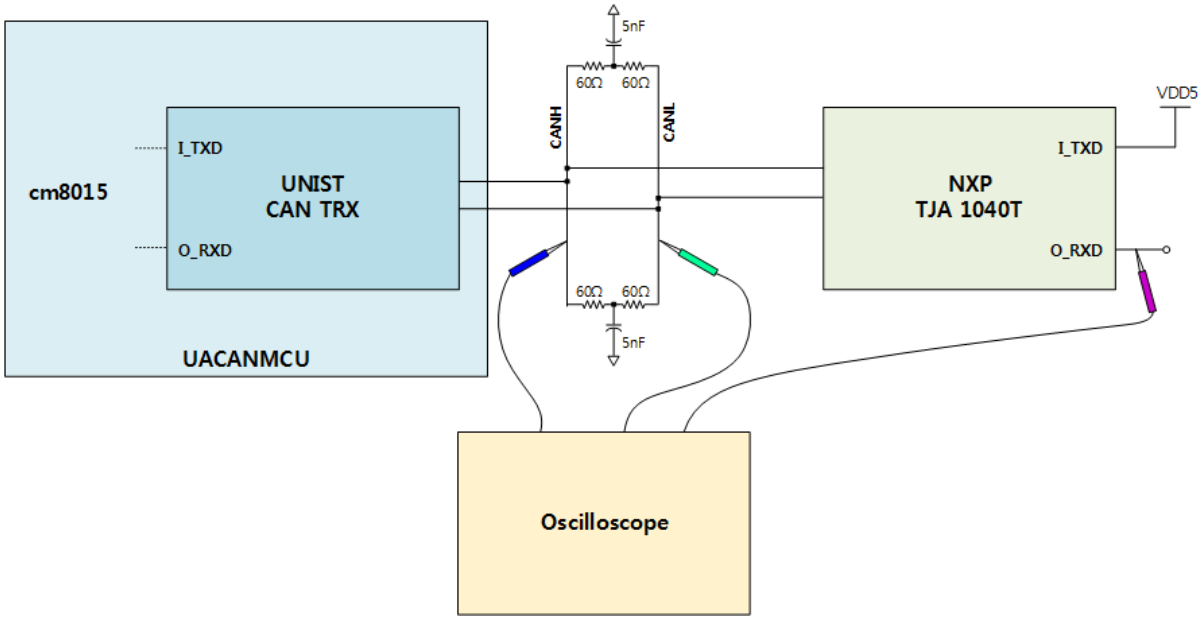
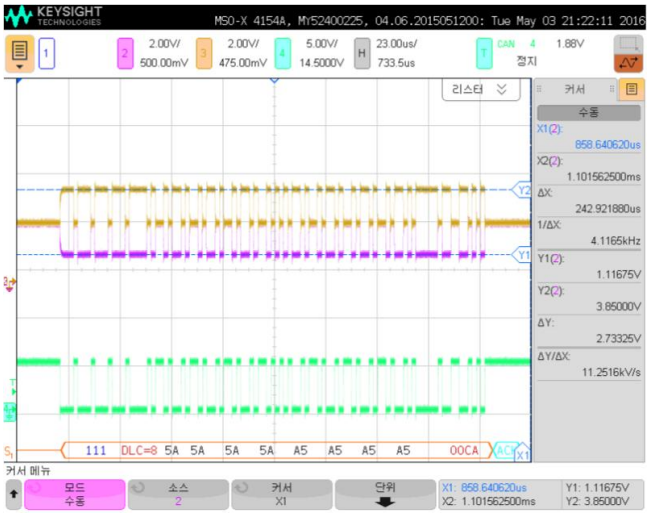
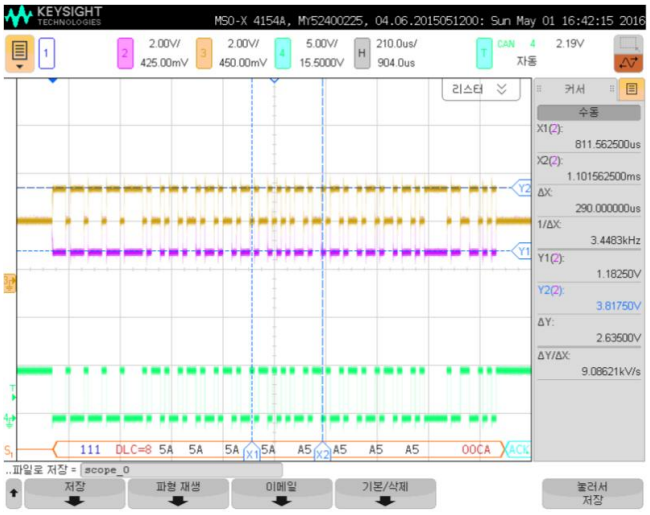


Figure 3.67 Test bed configuration for performance measurement (*Type II*)

(a) 1Mbps



(b) 500kbps



(c) 50kbps

Figure 3.68 Measurement results of *Type II*

IV. Interoperability Test

4.1 Interoperability Test with TI (Texas Instruments) Product

The designed CAN was tested with a commercial CAN transceiver (TI's SN65HVD231) [38]. The system for interoperability test was set as depicted in Fig. 4.1. ETRI's MECU was used for controlling the transceiver. It has five channels for CAN protocols, one of them was assigned for the transceiver of this work (CH#0 was assigned).

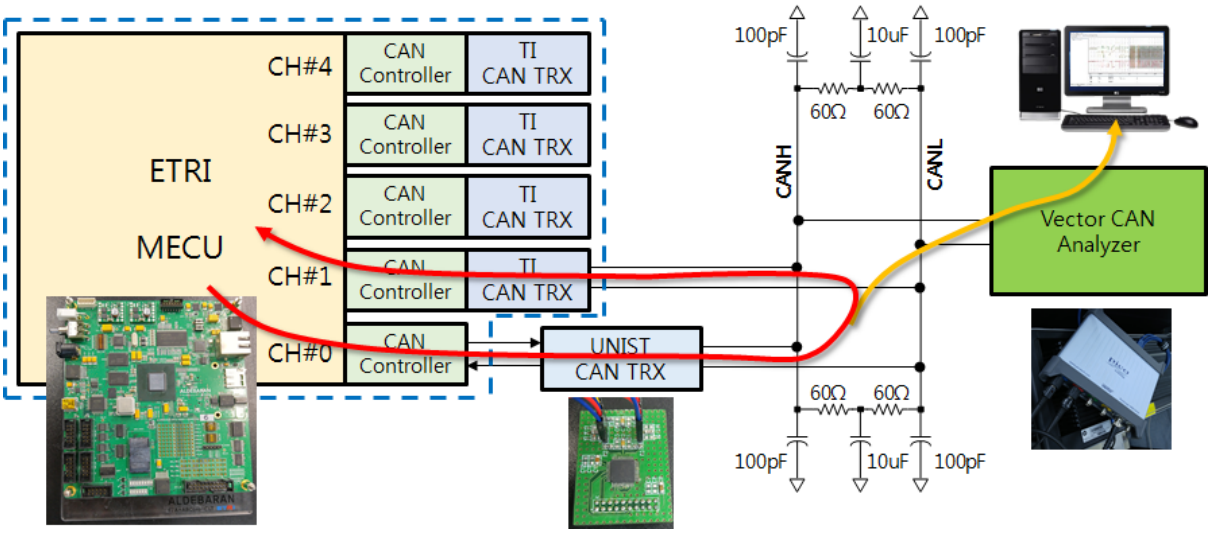


Figure 4.1 Configuration for interoperability test with TI CAN transceiver

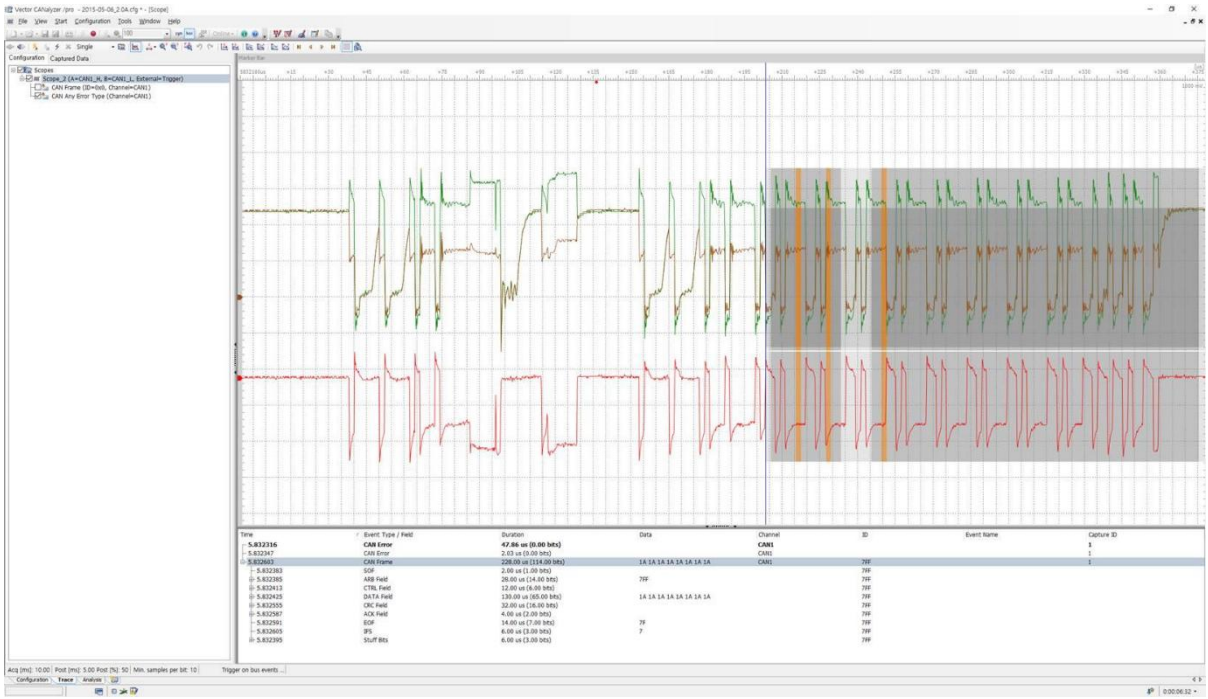


Figure 4.2 Interoperability test screen (measured by Vector CAN analyzer)

This test configuration shows a kind of loop-back test. Transceiver of CH#0 sends data, and CH#1 receives the data. Then, MECU checks if there is no error in data. The opposite directional communication is also possible. The communication status of CAN bus is recorded by Vector's CAN analyzer. As shown in Fig. 4.2, the bus signal rather noisy due to undesirable wiring and sub-optimized test PCB.

4.2 Demo system: Automotive Light Control System

Fig. 4.3 shows a block diagram of a demo system. The left side of the system is NODE#1. There is a MCU, AT90CAN128 [40]. It contains an integrated CAN controller for driving a CAN transceiver. NODE#1 senses user input via GPIO, and it transmits the control data to NODE#2 for activating the light system. NODE#2 receives the control data frame and switches the LEDs. The communication between NODE#1 and NODE#2 utilizes only CAN protocol.

There are two transceivers for communication; NXP PCA82C250 [32], [33] and ***Type I***. The transceiver is designed to be switchable, and can be switched by user. Therefore, there are four combinations for interoperability test. (1) ***Type I*** – ***Type I***, (2) ***Type I*** – NXP PCA82C250, (3) NXP PCA82C250 – ***Type I***, (4) NXP PCA82C250 – NXP PCA82C250.

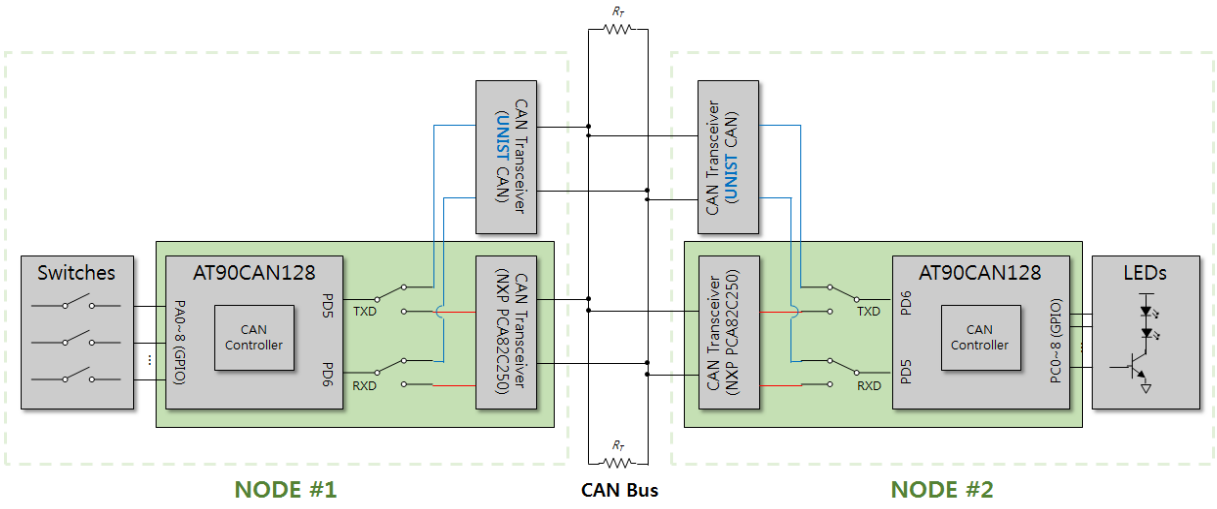


Figure 4.3 Block diagram of automotive light control system

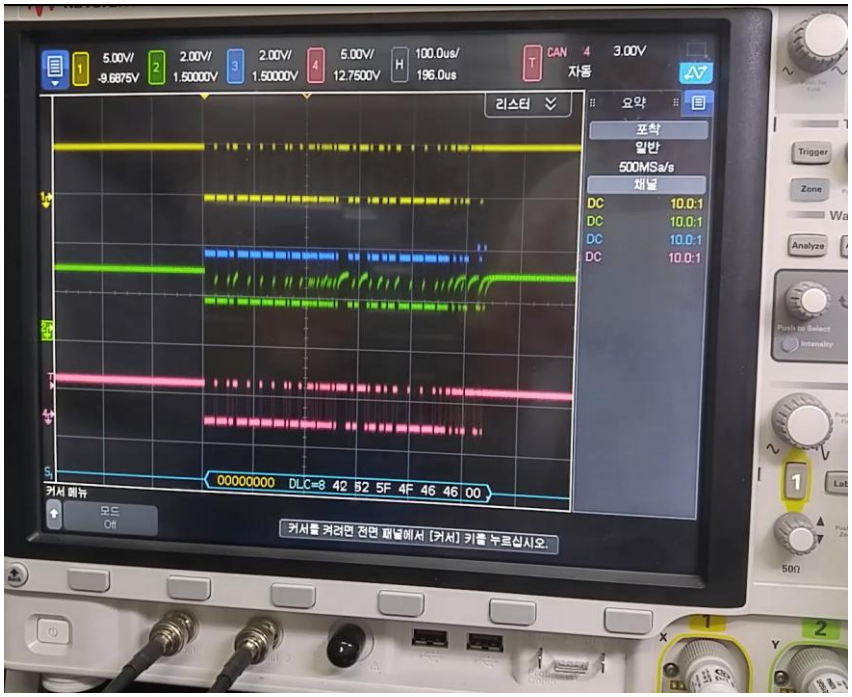


Figure 4.4 CAN communication monitoring (decoded by oscilloscope)

The communication status is monitored as shown in Fig. 4.4. The data rate was 125kbps. Fig. 4.5 shows the demonstration result that the modules are working properly.

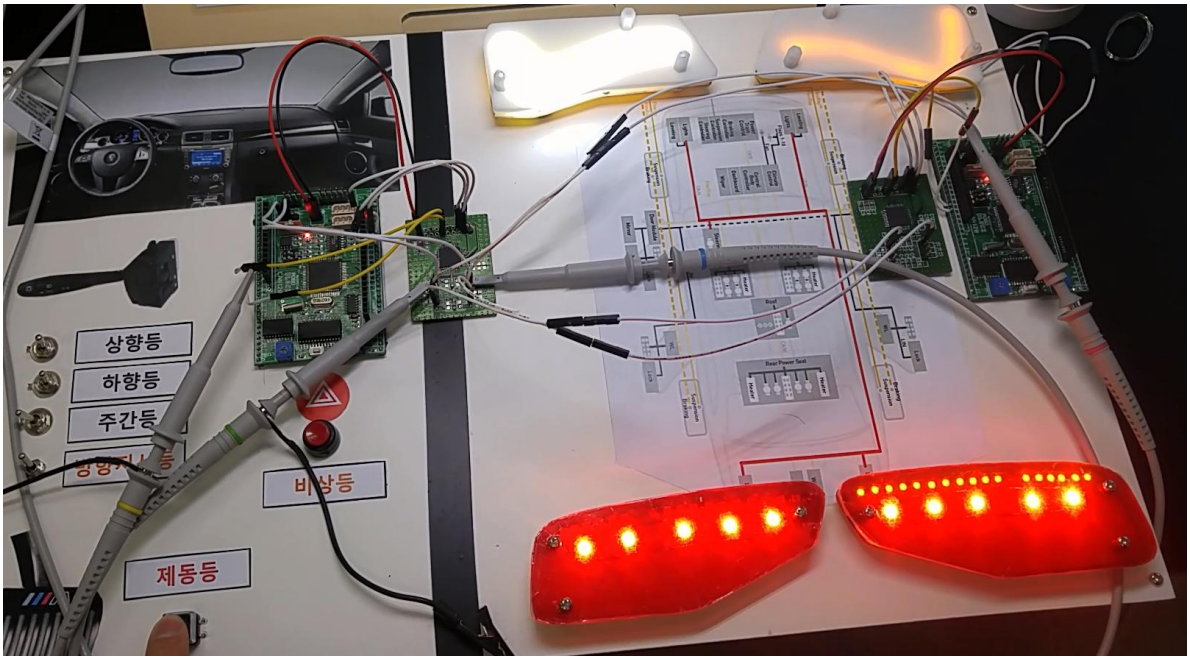


Figure 4.5 Performance demonstrations; right turn signal, day light, emergency blinker, brake light, and combination of them.

4.3 Demo System: Driver-side Door-zone Module

In this application, ***Type II*** was used. This application is DDM, the main functionality is motor control. It controls a side mirror of a car, for example, mirror folding/unfolding, and mirror position adjustment. CAN Node#1 senses user input via GPIO, and transmits control data frames to Node#2 by using TI's 3.3V CAN transceiver [38], [39]. Node#1 is implemented by MECU. Then, Node#2 receives the control data, and actuates the motor inside of the side mirror. Node#2 is implemented by Smart Output ASIC.

A large current of 3A (12V level) is required for driving the motor. However, the actuator driver of Smart Output ASIC is not realized yet. Thus, available output is only GPIO port which is 3.3V level with 20mA current output. The motor will be never moving with this GPIO output. Therefore, a relay module was employed for switching. Refer to Fig. 4.6. The gray box means the actuator driver which is not implemented yet. This relay module will be fully replaced by the Smart Output ASIC in near future, the next chip.

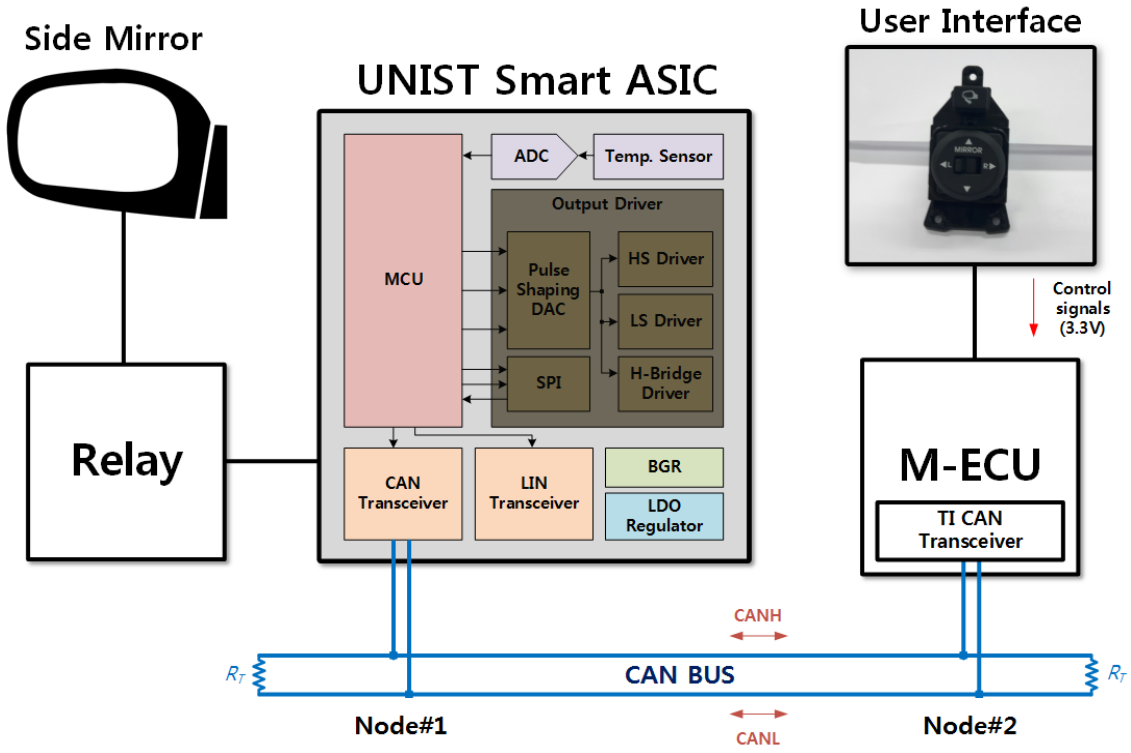


Figure 4.6 Block diagram of DDM demonstration system

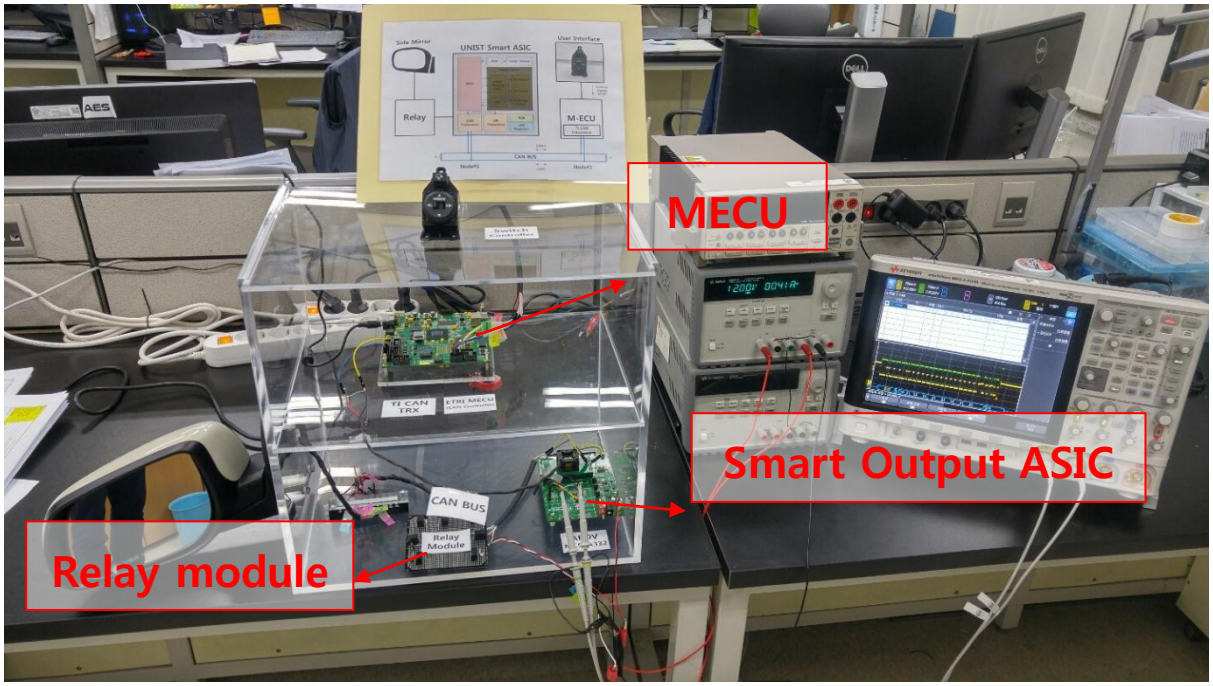


Figure 4.7 Assembled demonstration system of DDM

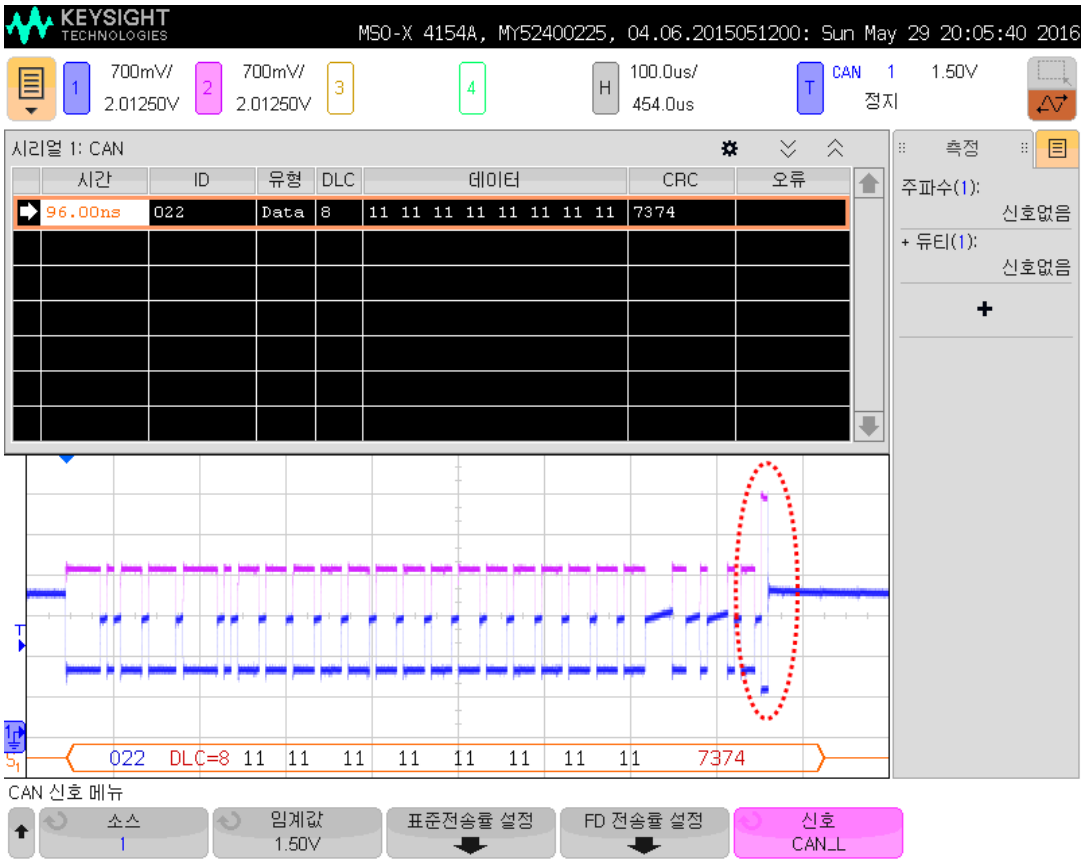


Figure 4.8 Communication between MECU and Smart Output ASIC (red circle: ACK signal)

Fig. 4.7 is the implemented system for DDM. Fig. 4.8 shows communication status. The CAN frame was decoded by oscilloscope. The data frame is transmitted by TI CAN transceiver of MECU. The red circle is an ACK signal came from Smart Output ASIC. The data rate was 125kbps. TI's CAN transceiver's output level is 3.3V, while the output level of ***Type II*** is 5V.

The demo system worked well. The functionality of the demo system is mirror folding/ unfolding, and 4-way mirror position adjusting.

V. Future Work

The CAN protocol has been and will be one of the main In-Vehicle-Network systems. Recently, CAN-FD protocol has been released to upgrade the performance of CAN interface [41]. Therefore, more research on the implementation on CAN-FD design is needed.

On the other hands, more research on a complete Smart Output ASIC should be carried on in order to show that this device would actually help to make the wiring harness system simpler and better as shown in Fig. 5.1.

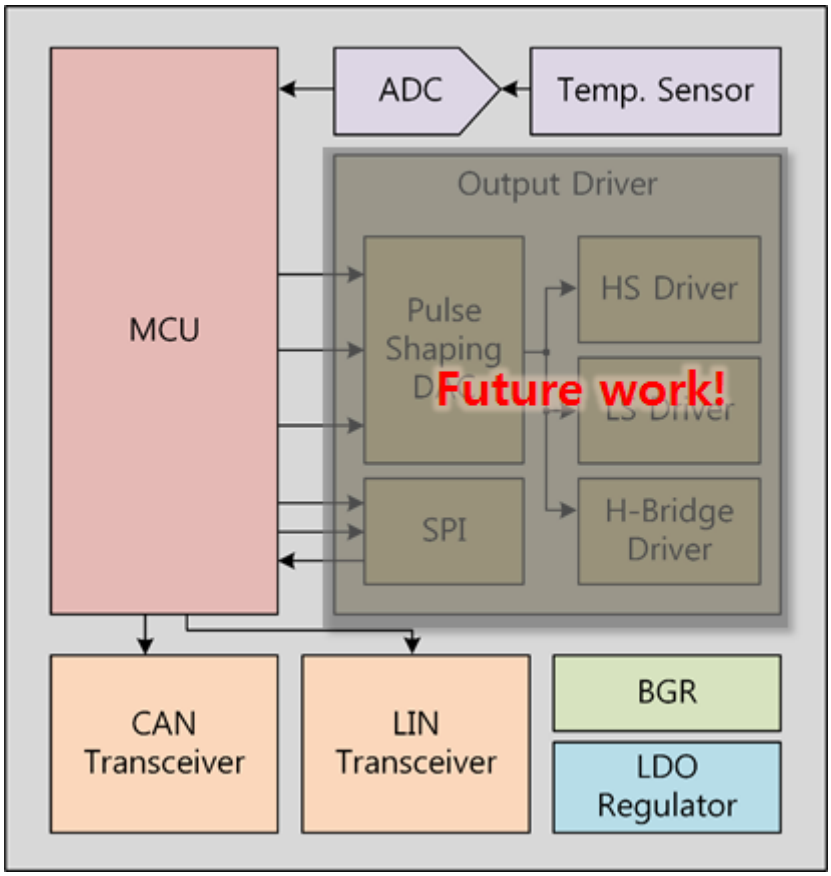


Figure 5.1 Future work for Smart Output ASIC

VI. Conclusion

Its full details of design, implementation, and measurement result of a CAN transceiver for Smart Output ASIC were presented. Two types of CAN transceiver were fabricated by using an automotive BCDMOS process. One is a stand-alone version (***Type I***), and another is a MCU integrated version (***Type II***) which is a mainframe of Smart Output ASIC. The size of implemented ***Type I*** core was 1,275um (W) × 1,125um (H). The size of ***Type II*** was 3,080um (W) × 3,680um (H). Its maximum data rate was 1Mbps, and it is proven in Section III.

As a part of application test, a test system was designed for each chip. One is automotive light control system which is using ***Type I***. There are commercial CAN transceivers in the application systems. In the automotive light control system, NXP PCA82C250 was used for performance comparison and interoperability test. The performance demonstrations were successful and working well. The results were presented in Section IV.

REFERENCES

[1] R. K. Jurgen, "Automotive electronics: Drivers get more options in 1983: Car makers embrace electronics in an all-out drive to improve performance, handling, fuel economy, riding quality, driver aids, and entertainment," *IEEE Spectrum*, vol. 19, no. 11, pp. 30-36, Nov. 1982.

[2] Andrew Chong, *Driving Asia: As Automotive Electronic Transforms a Region*, Infineon Technologies Asia Pacific Pte Limited, 2011.

[3] R. Charette, *This car runs on code*, Feb. 2009. [Online]: http://www.spectrum.ieee.org/feb09/7649

[4] *ISO 11898: Road Vehicles – Controller Area Network (CAN)*, International Organization for Standardization, Geneva, Switzerland, 2003.

[5] G. Leen and D. Heffernan, "Expanding automotive electronic systems," *IEEE Comput. Soc.*, vol. 35, no. 1, pp. 88–93, 2002.

[6] D. S. Eddy and D. R. Sparks, "Applications of MEMS technology in automotive sensors and actuators," *Proc. IEEE*, vol. 86, no. 8, pp. 1747–1755, Aug. 1998.

[7] M. R. Fairchild, R. B. Snyder, C. W. Berlin, and D. H. R. Sarma, "Emerging substrate technologies for harsh-environment automotive electronics applications," SAE Technical Paper Series 2002-01-1052.

[8] S. Franisse, *Application Requirements for Smart High-Side Switches*, Rev. 1.0, Infineon Technologies AG., Dec. 2010. [Online]: https://www.infineon.com

[9] Zhang Bin and Feng Quan-yuan, "A Novel thermal-shutdown Protection Circuit", *3rd International Conference on Anti-counterfeiting, Security, and Identification in Communication*, 2009.

[10] M. A. P. Pertijs, K. A. A. Makinwa, and J. H. Huijsing, "A CMOS smart temperature sensor with a 3σ inaccuracy of ±0.1 ◦C from −55 ◦C to 125 ◦C," *IEEE J. Solid-State Circuits*, vol. 40, no. 12, pp. 2805–2815, Dec. 2005.

[11] M. Glavanovics, H. Zitta, "Dynamic Hot Spot Temperature Sensing in Smart Power Switches," *ESSCIRC 2002*, Firenze, Italy.

[12] W. Horn and H. Zitta, "A robust smart power bandgap reference circuit for use in an automotive environment," *IEEE J. Solid-State Circuits*, vol. 37, no. 7, pp. 949 – 952, July 2002.

[13] S. Mitra, *Power-up Considerations (Technical note: AN522)*, Microchip Inc., 1997.

[14] M. Palmer, *Power-up Trouble Shooting (Technical note: AN607)*, Microchip Inc., 1997.

[15] Analog Devices Inc., *Power Supply Monitor with Reset (Technical note: ADM709)*, 1995.

[16] T. Yasuda and et al., "A Power-on Reset Pulse Generator for Low Voltage Applications," *IEEE ISCAS*, May 2001, vol. 4, pp. 599-601.

[17] Texas Instruments Inc., *LP3470 Tiny Power On Reset Circuit (Datasheet)*, Jun. 1999.

[18] Huy-Binh Le, "A Long Reset-Time Power-On Reset Circuit With Brown-Out Detection Capability," *IEEE TCAS-II*, vol. 58, no. 11, pp. 778-782, Nov. 2011.

[19] STMicroelectronics, *Low power reset circuit (Datasheet of STM1810)*, Sep. 2010.

[20] D. Burdia, "On-chip Power-on Reset Strategy and IO Power Supply Detection for VLSI Circuits," *IEEE ISSCS*, Jul. 2009, pp. 1-4.

[21] S. K. Wadhaw, "Zero Steady Current Power-on Circuit with Brown-Out Detector," *IEEE VLSID*, Jan. 2006, pp. 631-636.

[22] I. M. Filanovsky, "CMOS Schmitt Trigger Design," *IEEE TCAS-I*, vol. 41, no. 1, pp. 46-49, Jan. 1994.

[23] W. Prodanov, M. Valle, R. Buzas, and H. Piersciski, "Behavioral models of basic mixed-mode circuits: Practical issues and application," *in Proc. ECCTD*, Seville, Spain, Aug. 2007, vol. 1, pp. 854–857.

[24] W. Prodanov, M. Valle, R. Buzas, and H. Piersciski, "A mixed-mode behavioral model for a controller-area-network bus transceiver: A case study," *in Proc. IEEE BMAS*, San Jose, CA, Sep. 2007, pp. 67–71.

[25] A. J. Gines, E. Peralias, A. Rueda, N. M. Madrid, and R. Seepold, "A mixed-signal design reuse methodology based on parametric behavioural models with non-ideal effects," in Proc. DATE, Mar. 2002, pp. 310–314.

[26] R. Lissel and J. Gerlach, "Introducing new verification methods into a company's design flow: An industrial user's point of view," *in Proc. DATE*, Apr. 2007, pp. 1–6.

[27] W. Prodanov, M. Valle, and R. Buzas, "A controller area network bus transceiver behavioral model for network design and simulation," *IEEE Trans. Ind. Electron.*, vol. 56, no. 9, pp. 3762–3771, Sep. 2009.

[28] NXP Semiconductors, *High speed CAN transceiver (Datasheet of TJA1040)*, Feb. 2003.

[29] NXP Semiconductors, *AN10211: TJA1040 high speed CAN transceiver Rev.02 (Application note of TJA1040)*, Nov. 2006.

[30] NXP Semiconductors, *High-speed CAN transceiver Rev.04 (Datasheet of TJA1041A)*, Jul. 2008.

[31] NXP Semiconductors, *High-speed CAN transceiver with Standby mode Rev.7 (Datasheet of TJA1042)*, May 2012.

[32] NXP Semiconductors, *CAN controller interface Rev. 06 (Datasheet of PCA82C250)*, Aug. 2011.

[33] NXP Semiconductors, *AN96116: CAN Transceiver (Application note of PCA82C250)*, Oct. 1996.

[34] ON Semiconductors, *High Speed Low Power CAN Transceiver Rev.4 (Datasheet of NCV7341)*, May 2009.

[35] Infineon Technologies AG., *High speed CAN-Transceivers Rev. 1.20 (Datasheet of TLE8250G)*, Jul. 2013.

[36] Infineon Technologies AG., *High speed CAN Transceivers Application Note (Application note of TLE6250G)*, Jan. 2006.

[37] Freescale, *MC33901: High Speed CAN Transceiver Rev. 1.0 (Datasheet of MC33901)*, Dec. 2013.

[38] Texas Instruments Inc., *SLOS346H: 3.3-V CAN Transceiver (Datasheet of SN65HVD23x)*, Jul. 2006.

[39] Texas Instruments Inc., *SLOS346N: 3.3-V CAN Transceiver (Datasheet of SN65HVD23x)*, Jul. 2015.

[40] Atmel Corporation, *8-bit Microcontroller with 32K/64K128K Bytes of ISP Flash and CAN Controller (Datasheet of AT90CAN128)*, Aug. 2008.

[41] *CAN with Flexible Data-rate*, Robert Bosch GmbH, Aug. 2011.
[Online]: http://www.semiconductors.bosch.de

ACKNOWLEDGEMENT

I am highly honored and deeply grateful that I could get a time to do the research with my advisor, Dr. Myunghee Lee. He has supported my research work with all his effort. A great favor was granted to me. He always coached me with trust and encouragement. Thanks to his coach, I always achieved the amazing attainments which are impossible to accomplish if I do it alone. I appreciate Prof. Lee's teaching, lesson, and guidance about not only academic knowledge but also human life.

I would like to express my gratitude to Dr. Seokhyeong Kang. I appreciate his highly sophisticated class about digital circuit. Thanks to Prof. Kang's kind lesson, my knowledge about digital circuit were refined, and my design skill was also improved. His class was attractive and interesting. I felt a thirst for knowledge about digital circuit, and his class was 'antidipticum.'

For member of lab, Seokhwan, Youngwook, Kyunghee, Junho, Woojin, Jewoo, Wonkuk, Woojoong, and Yena, I've been happy during work together. I'm lucky to meet all of you, smart, talented and incisive. I couldn't make it without you. I realized a power of teamwork. Thank you, all of you guys and girl.

Yeonhwa and Hongkuk, thank you for your valuable helps. I always asked many things when I'm in the trouble, and you always helped me. Thank you for your professional advice and valuable counsel.

Special thanks to Dr. Changmin Kim, Myungsung Bae, Youngbae Jang of ABOV, and Dr. Youngsoo Kwon, Dr. Kyungjin Byun, and Dr. Nakwoong Uem of ETRI, and Sangwoo Kim of Crepas Technology.

Finally, I love you, father, mother, and my sister. I am here, thanks to you. Thank you for endless supports and love…